Notes on Mixed Integer Linear Programming

Participants of ISyE 6662

2019

Contents

Notation **v**

1 Formulations **1**
1.1 Mixed Integer Linear Programming 1
1.2 Some Pure IP Formulations 3
1.2.1 The Knapsack Problem 3
1.2.2 Covering, Packing, Partitioning 3
1.2.3 The Assignment Problem 4
1.2.4 Maximum Weighted Matching 4
1.2.5 Maximum Weight Forest 5
1.3 Modeling Disjunction 7
1.3.1 Motivating Example 7
1.3.2 Union of Polytopes 8
1.3.3 Piecewise Linear Functions 10
1.3.4 Log(n) Formulation for Sum of Binary Variables Equal to One 12
1.4 Traveling Salesman Problem 13
1.5 Lot-Sizing 15
1.6 Relaxation 16
1.7 Facility Location Problem 18
1.8 Suggested exercises 19

2 Introduction to Computational Complexity **39**
2.1 Standard encoding 39
2.2 Problem, instance, algorithm, worst-case running time 44
2.3 Theory of $\mathcal{NP}$-completeness 47
2.3.1 Introduction 47
2.3.2 $\mathcal{NP}$ and co-$\mathcal{NP}$ 48
2.3.3 $\mathcal{NP}$-Completeness 51

2.3.4 Binary integer program 54
2.3.5 3SAT 55
2.3.6 Vertex-Cover Problem 58
2.4 Suggested exercises 61

3 Review of Polyhedral Theory 63
3.1 Basic Definitions and Results 63
3.1.1 Fourier-Motzkin Projection 63
3.1.2 Duality Theory for Linear Programming 65
3.2 Recession Cone, Linearity Space of a Polyhedron 70
3.3 Dimension of a Polyhedron 71
3.4 Faces of a Polyhedron 73
3.5 Suggested exercises 76

4 Fundamental Theorem of IP 79
4.1 Fundamental Theorem of Integer Programming 79
4.2 Integral Polyhedron 80
4.3 A simple example 81
4.4 Proving a Polyhedron is integral 82
4.5 PORTA 83
4.5.1 Downloads 83
4.5.2 File Types 83
4.5.3 traf 84
4.5.4 fmel 85
4.6 Suggested exercises 86

5 TU Matrices and TDI Systems 91
5.1 Totally Unimodular Matrices and Their Applications in IP . 91
5.1.1 Hoffman and Kruskal Theorem 92
5.1.2 Testing for TU 94
5.2 Totally Dual Integral 97
5.2.1 Totally Dual Integral System (TDI) 98
5.3 Suggested exercises 99

6 Some Well-known Polytopes 103
6.1 Lot-sizing-problem 103
6.1.1 Lot-Sizing inequalities 104
6.1.2 The Separation Problem for Lot-Sizing Inequalities . . 105
6.1.3 Linear inequality description of the convex hull 106
6.2 Perfect Matching 110

6.3 Matroid and polymatroid . . . 115
6.3.1 Optimization over Matroids . . . 116
6.3.2 Matroid Polytope . . . 118
6.4 Suggested exercises . . . 120

7 Introduction to Cutting-plane Theory 123
7.1 Introduction . . . 123
7.1.1 Geometric Ideas . . . 123
7.1.2 Cuts from relaxation . . . 128
7.1.3 Algebraic approach . . . 130
7.1.4 Reformulation-Linearization Technique (RLT) . . . 133
7.2 Some Classical results on CG Cutting Planes . . . 133
7.3 Split Cuts for 0-1 IPs . . . 138
7.4 MIR, Gomory mixed integer (GMI) cut, Split cuts . . . 139
7.4.1 MIR is split cut . . . 139
7.4.2 A GMI cut is a MIR . . . 141
7.4.3 A split cut is a GMI cut . . . 142
7.5 Cover Inequality . . . 144
7.5.1 Knapsack Cover Inequality . . . 144
7.5.2 Flow Cover Inequality . . . 144
7.6 Lifting . . . 147
7.7 Suggested exercises . . . 151

8 Branch and Bound, Presolve, Primal Heuristics 155
8.1 Branch and Bound Algorithm . . . 155
8.1.1 Variable Selection . . . 155
8.1.2 Node Selection . . . 156
8.2 Preprocessing and Probing . . . 157
8.3 Primal Heuristics . . . 161

Notation

$\mathbb{R}^n$	Set of n-dimensional real vectors	
$\mathbb{R}^n_+$	Set of non-negative n-dimensional real vectors	
$\mathbb{Z}^n$	Set of n-dimensional integer vectors	
$\mathbb{Z}^n_+$	Set of non-negative n-dimensional integer vectors	
$[n]$	$\{1, \ldots, n\}$	(n is a natural number)
2^G	Power set of G	(G is a finite set)
$B(x, r)$	A closed ball around x or radius r	
$\delta(i)$	Neighbors of i	(i is a vertex of a graph)
$\text{conv}(S)$	Convex hull of a set S	
$\text{rec.cone}(S)$	Recession cone of a closed convex set S	
$\text{lin.space}(S)$	Lineality space of a closed convex set S	
$\text{extr}(P)$	Set of extreme points of P	
$\text{aff.hull}(F)$	Affine hull of F	
$\text{rel.int}(F)$	Relative interior of a convex set F	

Chapter 1

Mixed integer linear programming: Introduction and formulation

1.1 Mixed Integer Linear Programming

Let $\mathbb{R}^n$ be the n-dimensional Euclidean space. A general optimization problem deals with finding the "best solution" to a system of constraints, where best is with respect to a objective function.

Definition 1.1 (Optimization problem in general form)**.** *Let n and m be natural numbers. Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ for all $i \in \{1, \ldots, m\}$ be real-valued functions. Then a general optimization problem is of the form:*

$$\begin{array}{rll} \textit{sup} & g(x) & \\ \textit{such that} & f_i(x) \leq 0 & \forall i \in \{1, 2, ..., m\} \\ & x \in \mathbb{R}^n. & \end{array} \tag{1.1}$$

Henceforth we abbreviate 'such that' by '*s.t.*'. Note that we may substitute *sup* by *max* whenever optimal solutions of (1.1) exist.

Definition 1.2 (Mixed integer linear program (MILP))**.** *This is the class of optimization problems with linear objective, linear constraints and some variables are restricted to be integral, that is it is of the form:*

$$\begin{array}{rl} \max & c^T x \\ s.t. & Ax \leq b \\ & x \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}, \end{array} \tag{1.2}$$

where $\mathbb{Z}^{n_1}$ is the set of n_1-dimensional integer vectors.

Observation 1.1. *Notice that (1.2) is indeed a special form of (1.1). For example, if $x_i \in \mathbb{Z}$, then we can rewrite this as $sin(\pi x_i) \leq 0, -sin(\pi x_i) \leq 0$.*

We make a very important assumption through out for mixed integer programming model (1.2). We will assume unless otherwise stated that all data, that is c, A, b is rational. There are a number of reasons for making this assumption. We do not expect to typically obtain real life MILP instances with irrational data. Moreover, as can be seen in Homework 1 (also see Fundamental Theorem of Integer Programming) without the rationality assumption optimal solutions may not exist. Rationality of data assumption is crucial for many integer programming (IP) theorems to hold.

Definition 1.3 (Pure IP, Binary MILP, Linear program)**.**

1. *If $n_2 = 0$, then the model (1.2) is called a pure integer programming (IP) model.*
2. *If we have a constraint $0 \leq x_i \leq 1$ for every integer variable x_i, then the model is called a binary mixed integer program. That is, in this class of problems all the integer variables can take a value of either 0 or 1. This is one of the most prevalent classes of MILP models in real-world applications.*
3. *If $n_1 = 0$, then the model (1.2) reduces to an linear program (LP).*

A class of problem that can usually be modeled as binary IPs is combinatorial optimization problem. Below we describe a generic combinatorial optimization problem.

Definition 1.4 (Combinatorial optimization)**.** *Let G be a finite set, henceforth called the ground set. Let 2^G denote the power set of the ground set G. Let $c : G \to \mathbb{R}$. Then a typical combinatorial optimization problem is of the form:*

$$\begin{array}{ll} \max & \sum_{i \in b} c_i \\ s.t. & b \in B, \end{array}$$

where $B \subseteq 2^G$. The input to the problem is G, c and B and the task is to compute the optimal solution of the above problem.

Combinatorial problems have integer programming formulations.

1.2 Some Pure IP Formulations

In this section, we will formulate pure IP models for some classic problems.

1.2.1 The Knapsack Problem

The binary knapsack problem is as follows: given a set of n items, each with a utility u_i and weight w_i for $i \in \{1, \dots, n\}$, we must choose which items are to be packed into the knapsack (which is another word for 'backpack') such that we maximize the total utility, while ensuring that the total weight of the items chosen does not exceed the (carrying) capacity W. This is formulated as:

$$\begin{aligned} \max \ & \sum_{i=1}^{n} u_i x_i \\ \text{s.t.} \ & \sum_{i=1}^{n} w_i x_i \leq W \\ & x_i \in \{0,1\} \qquad \forall i \in \{1, \dots, n\} \end{aligned}$$

1.2.2 Covering, Packing, Partitioning

Let $G = \{1, 2, \dots, n\}$ and let $\{g_1, \dots, g_m\} = \mathcal{T} \subseteq 2^G$ where 2^G represents the power set of G.

Definition 1.5. *$S \subseteq \mathcal{T}$ is called a covering if*

$$\cup_{g \in S} g = G.$$

Definition 1.6. *$S \subseteq \mathcal{T}$ is called a packing if*

$$g_u \cap g_v = \emptyset \ \forall \ g_u, g_v \in S.$$

Definition 1.7. *$S \subseteq \mathcal{T}$ is called a partitioning if S is a covering and packing.*

Model. Let us construct a matrix $A \in \{0,1\}^{n \times m}$ where:

$$\begin{aligned} Data: \quad A_{ij} \quad &= \begin{cases} 1 & if\ i \in g_j \\ 0 & otherwise \end{cases} \\ Variables: \quad x_j \quad &= \begin{cases} 1 & if\ g_j \in S \\ 0 & otherwise \end{cases} \end{aligned}$$

These problems can be represented using the following model ($\mathbf{1}$ is a vector of all one's).

- Covering: $\{x \in \{0,1\}^m | Ax \geq \mathbf{1}\}$.
- Packing: $\{x \in \{0,1\}^m | Ax \leq \mathbf{1}\}$.
- Partitioning: $\{x \in \{0,1\}^m | Ax = \mathbf{1}\}$.

1.2.3 The Assignment Problem

Given a set of n machines and n operators, such that a profit of p_{ij} is obtained when operator i is assigned to machine j, the assignment problem is to find the best possible assignment of operators to machines such that each operator is assigned to only one machine and each machine is assigned to exactly one operator.
Here, the decision variable used is:

$$x_{ij} = \begin{cases} 1 & \text{if operator 'i' is assigned to machine 'j'} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i,j \in [n]$$

The formulation is as follows:

$$\begin{aligned} \max \ & \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}x_{ij} \\ \text{s.t.} \ & \sum_{i=1}^{n} x_{ij} = 1 && \forall j \in \{1,\dots,n\} \\ & \sum_{j=1}^{n} x_{ij} = 1 && \forall i \in \{1,\dots,n\} \\ & x_{ij} \in \{0,1\} \qquad \forall i \in \{1,\dots,n\} && \forall j \in \{1,\dots,n\} \end{aligned}$$

This is an example partitioning IP.

1.2.4 Maximum Weighted Matching

Definition 1.8 (Matching)**.** *Given a graph $G(V,E)$, a matching M is a subset of edges ($M \subseteq E$) such that no vertex is incident to more than one edge in the matching, i.e., the degree of each vertex in $G(V,M)$ is at most 1.*

The maximum weighted matching problem (to find a matching such that it maximizes the weight associated with it) is formulated below, where the decision variable is defined as:

$$x_e = \begin{cases} 1 & \text{if edge } e \text{ is in the matching} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } e \in E$$

and the model is:

$$\begin{aligned} \max \; & \sum_{e \in E} w_e x_e & & \\ \text{s.t.} \; & \sum_{e \in \delta(i)} x_e \leq 1 & \forall i \in V & \qquad (1.3) \\ & x_e \in \{0, 1\} & \forall e \in E & \qquad (1.4) \end{aligned}$$

where $\delta(i)$ represents the set of all edges that are incident to the vertex i. This is an example of packing type IP.
Remarks:

1. In the above formulation, if the inequality in (4.4) is replaced by $\sum_{e \in \delta(i)} x_e = 1$, then the problem becomes a perfect matching problem.

2. The assignment problem is a special case of perfect matching on a bipartite graph with $G = (M \cup W, E)$ and $|M| = |W| = n$, where M and W are the set of machines and operators respectively. G is a bipartite because every edge $e \in E$ has one end point in the set of machines and another in the set of operators.

1.2.5 Maximum Weight Forest

Definition 1.9 (Forest). *An acyclic graph is called a forest.*

Definition 1.10 (Tree). *An acyclic connected graph is called a tree.*

The maximum weighted forest problem is, given a graph $G = (V, E)$ and weights c_e $\forall e \in E$ associated with each edge, we must find a forest that maximizes the total weight. The decision variable is:

$$x_e = \begin{cases} 1 & \text{if edge } e \text{ is in the forest} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } e \in E$$

The problem is formulated as follows:

$$\max \sum_{e\in E} c_e x_e \tag{1.5}$$

$$\text{s.t.} \quad \sum_{e\in E(S)} x_e \leq |S| - 1 \qquad \forall S \subseteq V,\ |S| \geq 3 \tag{1.6}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{1.7}$$

where $E(S)$ is the set of edges where both vertices are in S. We will now show that the above formulation is correct for the maximum weighted forest problem.

Definition 1.11 (Degree of a vertex)**.** *The number of edges incident to a vertex.*

Lemma 1.1. *If F is a non-empty forest, then there is at least one vertex in F with degree equal to 1.*

Proof. Assume by contradiction, that every vertex of F has a degree of 2 or higher. Now, we pick any vertex v_1 and we can find a neighboring vertex, say v_2 and in this way, we can construct a path $v_1 - v_2 - v_3 - \cdots - v_k - \ldots$. Since the degree of every vertex is at least 2, we can always find a neighboring vertex until we reach a vertex v_i that has already been visited (since the number of vertices is finite). This implies that the graph is cyclic, which contradicts our assumption that F is forest. □

Proposition 1.1. *The constraints in (1.6) - (1.7) are satisfied for some x if and only if x is a forest.*

Proof. ($\Rightarrow$) Let $F \subseteq E$ not be a forest. We need to show that $\hat{x}$ corresponding to F does not satisfy the constraints (1.6) and (1.7). Assume, by contradiction, that $\hat{x}$ satisfies the constraints (1.6) and (1.7). Since F is not a forest then there is a cycle on the vertex set, say S. Then we have $\sum_{e\in E(S)} \hat{x}_e \geq |S|$, but this violates constraint (1.6) and we have a contradiction.

($\Leftarrow$) Let F be a forest. We need to show that x corresponding to F satisfies the constraints (1.6) and (1.7).

We prove this direction using induction on the size of the forest. Consider the base case where $n = 2$ where n is the size of the forest. Clearly if x corresponds to a forest on two nodes, it satisfies the constraints (1.6) and (1.7).

Assume that the statement holds for forests up to size N. We must show that it holds for forests of size $N+1$. From Lemma 1.1, we know that for a non-empty forest, there exists at least one vertex of degree 1. Let's call this vertex U. Constraint (1.7) holds. We now proceed by examining constraint (1.6) for different choices of S in the following cases:

1. $S \subseteq V,\ U \notin S$
   Also, $\sum_{e \in E(S)} x_e \leq |S| - 1$, from the induction hypothesis (the size of the forest is now 'N' because we have removed the isolated vertex) and hence, the statement holds.

2. $S \subseteq V,\ U \in S$
   $\sum_{e \in E(S)} x_e = \sum_{e \in E(S \setminus \{U\})} x_e + 1 \leq (|S| - 1 - 1) + 1 = |S| - 1$, where the inequality is implied by the induction hypothesis.

$\square$

1.3 Modeling Disjunction

1.3.1 Motivating Example

Consider (x, y) restricted to the feasible region of the union of the following two polytopes.

$$\begin{cases} x \leq a \\ x \geq 0 \\ y \leq d \\ y \geq 0 \end{cases} \quad or \quad \begin{cases} x \leq b \\ x \geq a \\ y \leq c \\ y \geq 0 \end{cases}$$

The union of the two polytopes could be modeled as the following. (Let $M = 2max\{a, b, c, d\}$)

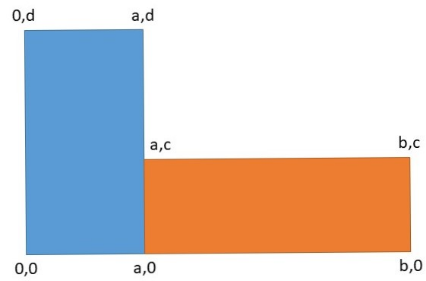


Figure 1.1: Union of two polytopes.

$$\begin{aligned}
x &\leq a + Mz^{*} \\
-x &\leq 0 + Mz^{*} \\
y &\leq d + Mz^{*} \\
-y &\leq 0 + Mz^{*} \\
x &\leq b + M(1-z)^{**} \\
-x &\leq -a + M(1-z)^{**} \\
y &\leq c + M(1-z)^{**} \\
-y &\leq 0 + M(1-z)^{**} \\
z &\in \{0,1\} \\
x, y &\in \mathbb{R}
\end{aligned}$$

Correctness. If $z = 1$, the constraints of the type $**$ become equivalent to the orange polyhedron while the other constrains become redundant. For example the first constraint of $x \leq a + Mz$ is impled by the constraint $x \leq b$ when $z = 1$. Similarly, if $z = 0$, the constraints of the type $*$ become equivalent to the blue polyhedron and the other constrains become redundant.

1.3.2 Union of Polytopes

First Method (Big M Method)

Let the feasible region be the union of t polytopes, $P^1, P^2, ..., P^t$ where $P^i := \{x \in \mathbb{R}^n | A^i x \leq b^i\}$ and A^i is a matrix of size $m_i \times n$.

Lemma 1.2. *There exists $M < \infty$ such that any $\hat{x} \in \cup_{j=1}^t P^j$ satisfies the system of inequalities $A^i \hat{x} \leq b^i + M\mathbf{1}$ for all $i = 1, 2, \ldots, t$.*

Proof. Clearly, the set $x \in \cup_{j=1}^t P^j$ is bounded, i.e. there exists $Q \in \mathbb{R}_+$ such that if $x \in \cup_{j=1}^t P^j$, then $||x|| \leq Q$. Let

$$B = max_{i \in \{1,\ldots t\}, j \in \{1,\ldots,m_i\}} \{|b_j^i|\}$$
$$T = max_{i \in \{1,\ldots t\}, j \in \{1,\ldots,m_i\}} \{\|A_j^i\|\},$$

where A_j^i is the j^{th} row of A^i and b_j^i is the j^{th} entry of b^i. Set $M := B + TQ$. Now let $\hat{x} \in P^i$ for some $i \in \{1, ..., t\}$ and examine the j^{th} constraint of the system $A^k x \leq b^k + M1$:

$$(A_j^k)^T \hat{x} - b_j^k \leq |(A_j^k)^T \hat{x}| + |b_j^k| \leq |A_j^k||\hat{x}| + |b_j^k| \leq TQ + B = M.$$

Thus any $\hat{x} \in \cup_{j=1}^t P^j$ satisfies the system of inequalities $A^i \hat{x} \leq b^i + M1$ for all $i = 1, 2, \ldots, t$. □

Now, we present the general model of the first method:

$$\begin{aligned} A^i x &\leq b^i + M\mathbf{1}(1 - z^i) \quad i = 1, ..., t \\ \sum_i z^i &= 1 \\ z^i &\in \{0, 1\} \quad i = 1, ..., t \\ x &\in \mathbb{R}^n \end{aligned}$$

where M satisfies the property mentioned in Lemma 1.2, i.e. any $\hat{x} \in \cup_{j=1}^t P^j$ satisfies the system of inequalities $A^i \hat{x} \leq b^i + M\mathbf{1}$ for all $i = 1, 2, \ldots, t$.

Proposition 1.2. *The above model is correct.*

Proof. Let $x \in P^j$. We show that there exists z such that the above model is satisfied . Set $z^j := 1$ and $z^i := 0$ for all $i \neq j$. Then, we will have $A^j x \leq b^j$ and $A^i x \leq b^i + M1 \quad \forall i \neq j$ which holds due to our choice of M.

On the other hand, if $x \notin P^i \quad \forall i = 1, \ldots, t$, i.e., if x is not in any of the polytopes, then we show that there is no z such that (x, z) satisfies the constraints in the model. Assume by contradiction, that there exists a z such that (x, z) satisfies the constraints. Then $z \in \{0, 1\}^n$ such that $z_i = 1$ for some i and 0 otherwise. Then, the inequality $A^i x \leq b^i$ is enforced. However, x does not satisfy any of the constrains which is a contradiction. Hence, the model is correct. □

Second Method

Lemma 1.3. *Let $\{x \mid Ax \leq b\}$ is feasible and bounded. Then $Ax \leq 0$ is satisfied only for $x = 0$.*

Proof. Clearly 0 satisfies $Ax \leq 0$. No assume by contradiction, $Au \leq 0$ where $u \neq 0$. Consider $v \in \{x \mid Ax \leq b\}$. Then note that $A(v + \lambda u) \leq b + \lambda Au \leq b$ for $\lambda \geq 0$. Thus $v + \lambda u \in \{x \mid Ax \leq b\}$ for arbitrary large values of λ. This $\{x \mid Ax \leq b\}$ is not bounded, a contradiction. □

Here is the model of the second method:

$$\begin{aligned} A^i x^i \leq b^i z^i \ \forall i \in [t] \\ x = \sum_{i=1}^{t} x^i \\ \sum_{i=1}^{t} z^i = 1 \\ z^i \in \{0, 1\}, \quad i = 1, \dots, t. \end{aligned}$$

Proposition 1.3. *The above model is correct.*

Proof. Let $\hat{x} \in P^j$. We show that there exists $(x^1, \dots, x^t, z)$ such that the above model is satisfied. Set $z^i := 0 \quad \forall i \neq j$, $x^j := \hat{x}$, and $x^i := 0 \quad \forall i \neq j$. Then, the constraints of the model are satisfied.

If $x \notin P^i \quad i = 1, \dots, t$, i.e., if x is not in any of the polytopes, then we show that there are no $(x^1, \dots, x^t, z)$ such that the above constrains are satisfied. Assume by contradiction that there exists $(x^1, \dots, x^t, z)$ that satisfies the constraints. Then $z_j = 1$ for some j and 0 otherwise. Then, by Lemma 1.3 the only solution to $A^i x^i \leq 0 \quad \forall i \neq j$ is $x^i = 0$. Thus, we must have that $x = x^j$. However, $A^j x \leq b^j$ is not satisfied which is a contradiction. Hence, the model is correct. □

1.3.3 Piecewise Linear Functions

Definition 1.12. *Let $-\infty < a_1 < a_2 ... < a_n < \infty$. A function of the form $f : [a_1, a_n] \to \mathbb{R}^1$ is a piecewise linear continuous function if*

$$f(x) = m_i x + c_i \quad x \in [a_i, a_{i+1}] \quad \forall i \in \{1, ..., n-1\}$$

such that $m_i a_{i+1} + c_i = m_{i+1} a_{i+1} + c_{i+1} \quad \forall i \in \{1, ..., n-2\}$

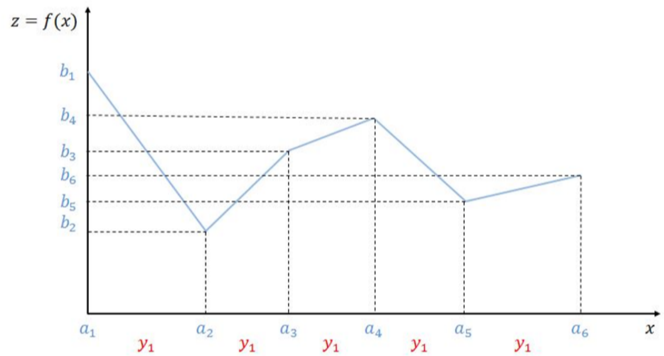


Figure 1.2: Example of piecewise linear function.

Observations.

- A function of the form $f(x_1, ..., x_p) = \sum_{j=1}^{p} f_j(x_j)$ is a piecewise linear function if $f_j(x_j)$ is piecewise linear for each j.
- An arbitrary continuous function of one variable can be approximated by a piecewise linear function, with the quality of the approximation controlled by the size of the linear segments.
- Note that modelling a piecewise linear function can be viewed as union of polytopes:

$$\bigcup_{j \in t} \text{conv}\{(x, y) \,|\, x = a_i + \lambda(a_{i+1} - a_i), y = m_i x + c_i, \lambda \in [0, 1]\}.$$

Next we present a slightly different model.

Suppose we have a piecewise linear function $f(x)$ specified by the points $(a_i, b_i = f(a_i))$ for $i = 1, ..., 6$. Then any $a_1 \le x \le a_6$ can be written as:

$$x = \sum_{i=1}^{6} w_i a_i, \quad \sum_{i=1}^{6} w_i = 1, \quad w \in \mathbb{R}_+^6.$$

If $a_i \le x \le a_{i+1}$ and w is chosen so that $x = w_i a_i + w_{i+1} a_{i+1}$ and $w_i + w_{i+1} = 1$, then we obtain $f(x) = w_i f(a_i) + w_{i+1} f(a_{i+1})$. In other words:

$$f(x) = \sum_{i=1}^{6} w_i f(a_i), \quad \sum_{i=1}^{6} w_i = 1, \quad w \in \mathbb{R}_+^6.$$

Therefore the requirement is at most two of the $w_i's$ are positive, and more over say w_j and w_k are positive, then $k = j - 1$ or $k = j + 1$. This condition can be modeled using binary variables $y_i \ for \ i = 1, ..., 5$ where $y_i = 1 \ if \ a_i \leq x \leq a_{i+1}$ and $y_i = 0$ otherwise. The additional constraints therefore are:

$$\begin{aligned} w_1 &\leq y_1 \\ w_i &\leq y_{i-1} + y_i \quad for \ i = 2, \ldots, 5 \\ w_6 &\leq y_5 \\ \sum_{i=1}^{5} y_i &= 1 \\ y_i &\in \{0, 1\} \quad for \ i = 1, \ldots, 5. \end{aligned}$$

1.3.4 Log(n) Formulation for Sum of Binary Variables Equal to One

Often in formulation of the MIP problems (such as disjunction and piecewise linear functions discussed above), we have binary constrains of the form:

$$\begin{aligned} \sum_{i=1}^{n} z_i &= 1 \\ z &\in \{0, 1\}^n \end{aligned}$$

These constraints could be modeled with $\log(n)$ order number of binary variables and the constraints as shown below.

$$\begin{aligned} \sum_{j|j \neq i} z_j &\leq \sum_{k|k \in supp(B(i))} (1 - w_k) + \sum_{k|k \notin supp(B(i))} w_k \quad \forall i = \{1, ..., n\} (*) \\ \sum_{i=1}^{n} z_i &= 1 \\ w &\in \{0, 1\}^{\lceil log_2(n+1)+1 \rceil} \\ z &\in \mathbb{R}_+^n \end{aligned}$$

where $\text{supp}(B(i))$ represents the positions of 1 in the binary number of i. For example, $\text{supp}(B(9))$ is $\{1, 4\}$ since binary number of 9 is 1001, and the positions of '1' are at 1 and 4. Note that the z_i's are not binary, and therefore, we have reduced the number of binary variables.

In this formulation, the w_k variables determine a single number between 1 and n, say t. Then, the $(*)$ constraint for $i = t$ will set all the z_i's other than z_t to 0. Since we have the constraint $\sum_{i=1}^{n} z_i = 1$, then z_t becomes 1. Note that, z_i will not be forced to be set to 0 on the other $(*)$ constraints since at each of those constraints at least one of the sums on the RHS of the equation will be nonzero.

1.4 Traveling Salesman Problem

In this problem, we assume n cities and d_{ij} is the distance for travel from i and j. Let

$$x_{ij} = \begin{cases} 1 & \text{if salesman travels from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } i, j \in [n], i \neq j.$$

First Formulation. The problem is formulated as follows.

$$\begin{array}{lrr} min & \sum_{i,j} d_{ij} x_{ij} & (1.8) \\ st. & \sum_i x_{ij} = 1 \quad \forall j = \{1, ..., n\}(*) & (1.9) \\ & \sum_j x_{ij} = 1 \quad \forall i = \{1, ..., n\}(**) & (1.10) \\ & \sum_{i \in U, j \in V \setminus U} x_{ij} \geq 1 \quad \forall\ U \subseteq V \ \textit{such that } 2 \leq |U| \leq n-1(***) & \\ & x_{ij} \in \{0, 1\} & (1.11) \end{array}$$

Second Formulation. Constraint (***) in this formulation prevents subtours. These constrains could alternatively be modeled as follows. Define u_i $i = 2, ..., n$ and replace the (***) constraints with:

$$u_i - u_j + n x_{ij} \leq n - 1 \quad \forall i, j \in \{2, ..., n\} \quad \oplus$$

as an alternative formulation.

In order to verify the alternative formulation, see that if a binary vector x that does not represent a tour while satisfying (*) and (**), then x represents at least two subtours, one of which does not contain node 1. By summing $\oplus$ over the arc set of some subtour that does not contain node 1, we obtain

$$\sum_{ij \in A'} x_{ij} \leq |A^{'}|(1 - \frac{1}{n})$$

Therefore, $\oplus$ excludes all subtours that do not contain node 1 and as a result, excludes all solutions that contain subtours.

On the other hand, no tours are excluded by $\oplus$ since for any tour there exists a corresponding u satisfying the constraint. In particular, set $u_i = k$, where k represents the position of node i in the corresponding tour. If $x_{ij} = 0$, then $u_i - u_j + nx_{ij} \leq n - 2$ holds and if $x_{ij} = 1, u_i = k$ and $u_j = k + 1$ for some k, then $u_i - u_j + nx_{ij} \leq n - 1$ is satisfied. Therefore, the alternative formulation is correct.

Third Formulation. To enforce connectivity, we introduce a new variable and corresponding constraints. Assume that we inject $n - 1$ units of goods into node 1 and take out 1 unit of goods at the following nodes $2, \ldots, n$, the system should be well balanced and connected.
New variable: (flow variable)

$$f_{ij} = \textit{flow along the edge } (i, j) \tag{1.12}$$

New constraint:

$$f_{ij} \leq (n-1)x_{ij} \forall i, j \in [n] \tag{1.13}$$

$$\sum_{j:j\neq 1} f_{j1} + n - 1 = \sum_{k:k\neq 1} f_{1k} \tag{1.14}$$

$$\sum_{j:j\neq i} f_{ji} - \sum_{k:k\neq i} f_{ik} = 1, \quad \forall i \in \{2, \ldots, n\} \tag{1.15}$$

$$f_{ij} \geq 0, \quad \forall (i, j) \in E \tag{1.16}$$

Proposition 1.4. *The formulation given by (1.8), (1.9), (1.10), (1.11), (1.13), (1.14), (1.15), (1.16) is a correct formulation of TSP.*

Proof. $\Leftarrow$ First, we prove that there is no subtour in a solution that satisfies the constraints of the model. Assume by contradiction that $(\hat{x}, \hat{f})$ satisfies the constraints but contains a subtour. Thus there exists a subtour that does not include the node 1. Let this subtour be $(i_1, i_2, \ldots, i_k)$. Add the constraints (1.15) for nodes in this sub tour. We obtain $0 = k$, a contradiction. Therefore, a feasible solution of the model contains no subtour.

$\Rightarrow$ Second, we show that given a tour, there always exist values for the flow variable such that the constraints of the model are satisfied. This can be easily verified as follows: Let the tour be $1, i_1, i_2, \ldots, i_{n-1}, 1$. Then set $f_{1,i_1} = n - 1$, $f_{i_1,i_2} = n - 2, \ldots, f_{i_k,i_{k+1}} = n - k + 1, \ldots, f_{i_{n-2},i_{n-1}} = 1$ and $f_{ij} = 0$ otherwise. It is easily verified that with these values of the flow variables, the constraints (1.8), (1.9), (1.10), (1.13), (1.14), (1.15), (1.16) are satisfied. Therefore, a tour is always feasible for the system. $\square$

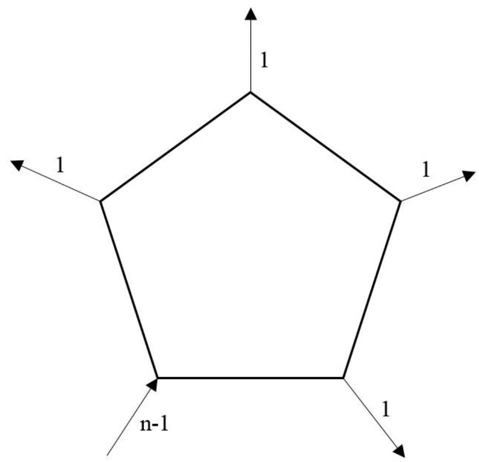


Figure 1.3: TSP: flow variable modelling

1.5 Lot-Sizing

In this problem, at each period a demand needs to be met (the demand is known in advance). At each period, the manufacturer decides how much to produce, and there is a fixed cost if production occurs at a period. A holding cost is also incurred for inventory that is kept for the next periods. The problem is to satisfy the demand while minimizing costs which include production costs at each period, fixed cost of production at each period, and inventory cost between the periods. The following are the model parameters:

- Time index: $1, \ldots, n$
- Demand: $\{d_1, \ldots, d_n\}$
- Cost per unit of production at each period: $\{c_1, \ldots, c_n\}$
- Cost per unit of inventory storage: $\{h_1, \ldots, h_n\}$
- Fixed cost in each time period if production is positive: $\{f_1, \ldots, f_n\}$

The variables of the model include:

- x_i: quantity produced at period i
- s_i: quantity stored from period i to period $i+1$
- y_i: binary variable equal to 1 if x_i is positive and 0 otherwise.

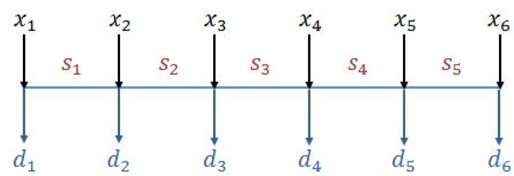


Figure 1.4: Lot-sizing model variables.

The optimization model for lot-sizing problem can be written as:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} f_i y_i + \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n-1} h_i s_i \\
\text{s.t.} \quad & x_1 = s_1 + d_1 \\
& s_{i-1} + x_{i-1} = s_i + d_i \quad for\ i = 2, \ldots, n-1 (*) \\
& s_{n-1} + x_n = d_n \\
& x_i \leq (\sum_{i=1}^{n} d_i) y_i \quad for\ i = 1, \ldots, n (**) \\
& y_i \in \{0, 1\} \quad for\ i = 1, ..., n
\end{aligned}
$$

(*) corresponds to the flow conservation constraints and (**) to the fixed charge constraints.

1.6 Relaxation

As we shall see, it is possible to construct multiple formulations. How do we compare these formulations? One approach is to understand the notion of relaxation.

A feasible solution to a maximization problem (resp. minimization problem) yields a lower (resp. upper) bound on the optimal objective function value. We now present a general technique to produce bounds in the other direction on the optimal objective value. These bounds are usually called as dual bounds.

Let P be the optimization problem

$$z^* := sup\ f(x)$$
$$x \in S$$

And let R be the optimization problem

$$w^* := sup\ g(x)$$
$$x \in \tilde{S}$$

Definition 1.13. *The optimization problem R is called the relaxation of the optimization problem P if*

- $S \subseteq \tilde{S}$
- $\forall x \in S, f(x) \leq g(x)$.

Proposition 1.5. *Let R be a relaxation of P, where R, P, z^*, w^* are defined as above, then*

$$w^* \geq z^*$$

Proof. If $S = \emptyset$, then we have by convention $sup_{x \in S} f(x) = -\infty$. Thus, there is nothing to prove in this case, so we assume $S \neq \emptyset$. Let $\epsilon > 0$. We need to show that $w^* \geq z^* - \epsilon$. By definition of z^*, there exists $x_\epsilon \in S$ such that $f(x_\epsilon) \geq z^* - \epsilon$. By definition of R, we have that $g(x_\epsilon) \geq f(x_\epsilon)$ or equivalently $g(x_\epsilon) \geq z^* - \epsilon$. Finally, since $S \subseteq \tilde{S}$, we have that $x_\epsilon \in \tilde{S}$ and therefore we obtain that $w^* \geq z^* - \epsilon$. □

Thus relaxations is a fundamental technique to judge the quality of a given feasible solution: For example, if we are solving a maximization problem and the objective function value is 10. On the other hand suppose we are able to construct a relaxation whose objective function value is 12. Then we know that our feasible is "pretty decent", i.e. it is within $\frac{12-10}{12} \times 100 \equiv 16.67\%$ of optimal solution.

It is clear that many types of relaxations may be used. Ideally, a relaxation should be computational tractable and also provide good dual bounds. In most IP solvers, the so-called linear programming relaxation is used. We present the formal definition next.

Definition 1.14. *Given the mixed integer program*

$$\begin{aligned} max \quad & c^T x \\ & Ax \leq b \\ & x \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \end{aligned}$$

we call the linear program

$$\begin{aligned} max \quad & c^T x \\ & Ax \leq b \\ & x \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \end{aligned}$$

as its linear programming relaxation (LPR).

Comparing Formulations Often the same MIP feasible set may be modeled using different choice of variables and optionally with the addition of some extra variables. As discussed before, one theoretical way to compare quality of two formulations is to compare their linear programming relaxations. Suppose we have two models for the same MIP set: Model 1 and Model 2. We say that Model 1 is better than Model 2, if for all objective functions, the LPR of Model 1 gives better bounds that LPR of Model 2. This happens when the feasible region of LPR of Model 1 is contained in the LPR of Model 2. Model 1 is better than Model 2 implies that LPR of Model 1 gives a better dual bound than the LPR of Model 2.

1.7 Facility Location Problem

Let demand be $D := \{1, \ldots, n\}$. Potential facilities $:= \{1, \ldots, m\}$. Define:

$$x_j = \begin{cases} 1 & \text{if facility } j \text{ is opened} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } j \in [m].$$

Let y_{ij} = percentage of demand of i satisfied by facility j, for $i \in [n], j \in [m]$.

We consider two possible formulations of this problem.

First formulation.

$$\begin{aligned}
min \quad & \sum_{j=1}^{m} c_j x_j + \sum_{i,j} h_{ij} y_{ij} \\
st. \quad & \sum_{j=1}^{m} y_{ij} = 1 \quad \forall i \in [n] \\
& y_{ij} \leq x_j \quad \forall i \in [m], j \in [n](*) \\
& x_j \in \{0,1\} \quad \forall j \in [m] \\
& y_{ij} \geq 0 \quad \forall i \in [n], j \in [m]
\end{aligned}$$

Second formulation.

$$\begin{aligned}
min \quad & \sum_{j=1}^{m} c_j x_j + \sum_{i,j} h_{ij} y_{ij} \\
st. \quad & \sum_{j=1}^{m} y_{ij} = 1 \quad \forall i \in [n] \\
& \sum_{i=1}^{n} y_{ij} \leq n x_j \quad \forall j \in [m](**) \\
& x_j \in \{0,1\} \quad \forall j \in [m] \\
& y_{ij} \geq 0 \quad \forall i \in [n], j \in [m]
\end{aligned}$$

The LPR of the first formulation is smaller than the LPR of the second formulation. The underlying reason is that the constraints of type (**) is the summation of constraints of type (*) over $i = 1, ..., n$ and that the summation of constraints generate a bigger feasible domain. Therefore, the first formulation is better even though it has more constraints.

Moreover, the containment of the LPRs could be strict as illustrated by the following example: Take $n = 2$ and $m = 3$, i.e. two demand nodes and 3 facilities. One can easily check that the solution:
$y_{11} = 1/2$, $y_{12} = 1/2$, $y_{13} = 0$, $y_{21} = 0$, $y_{22} = 1/2$, $y_{23} = 1/2$, $x_1 = 1/4$, $x_2 = 1/2$, $x_3 = 1/4$, satisfies the LPR of second formulation but not the first.

1.8 Suggested exercises

1. (Importance of rational data) Problem 1, Page 22, Textbook.

2. (Wolsey) uppose that you are interested in choosing a set of investments $\{1, \dots, 7\}$. Assume that they have a total cost of \$5, \$7, \$6, \$3, \$9, \$12, \$5. Assume there is a budget of \$30. Also assume that the eventual expected profit out of each investment is \$1, \$3, \$2, \$4, \$1, \$5, \$4. Write a IP formulation for maximizing profit. Now add constraints to model the following:

   - You must choose at least one of them.
   - Investment 1 cannot be chosen if investment 3 is chosen.
   - Investment 4 can be chosen only if investment 2 is also chosen.
   - You must choose either both investments 1 and 5 or neither.
   - You must choose either at least one of the investments 1,2,3 or at least two investment from 2,4,5,6.

3. A machine tool plant owns four different machines on which it can process jobs. If a machine is used at all, then a setup time is needed. A job cannot be divided between machines, that is each job must be processed by exactly one machine. (Note however that one machine can process more than one job). The relevant times in minutes are given in the following table.

	Job 1	Job 2	Job 3	Job 4	Machine Setup
Machine 1	42	70	930	710	10
Machine 2	340	43	120	7	20
Machine 3	560	32	40	9	60
Machine 4	71	760	5	80	85

   (a) The company's goal is to minimize the sum of the setup and machine operation times needed to complete all jobs. Formulate this problem as an integer linear program.

   (b) Assume that we want to impose that no more than two jobs can be assigned for any given machine. What constraints should be added to your model to represent this new restriction? (Just write constraint(s) for this part.)

   (c) Assume that we want to impose the restriction that if machine 1 is used then machine 3 is used. What constraint should be added to your model to represent this new restriction? (Just write constraint(s) for this part.)

(d) Assume that we want to impose the restriction that either machine 2 or machine 4 is used. What constraint should be added to your model to represent this new restriction? (Just write constraint(s) for this part.)

(e) Assume that we want to impose the restriction that: If machine 1 and machine 3 are both set up, then job 1 should not be assigned to machine 3. (Just write constraint(s) for this part.)

4. To graduate from ABC University with a major in OR, a student must complete at least two math courses, at least two OR courses, and atleast two computer courses. Some courses can be used to fulfill more than one requirement.

   (a) Calculus can fulfill the math requirement;
   (b) Optimization, math and OR requirements;
   (c) Data structures, computer and math requirements;
   (d) Computer simulation, OR and computer requirement;
   (e) Introduction to computer programming, computer requirement;
   (f) Forecasting, OR and math requirement; and
   (g) Business Statistics, OR requirement.

   Some courses are pre requisites for others:

   (a) Calculus is a prerequisite for Business Statistics.
   (b) Introduction to computer programming is a prerequisite for Computer simulation and Data structures
   (c) Business Statistics is a prerequisite for Forecasting.

   Formulate an integer program to minimize the number of courses needed to satisfy the major requirements.

5. Show how to model the following situations. You may create integer variables when necessary. You may assume that both x_1 and x_2 are between 0 and 10. If you use Big-M values be sure to clearly state what the value of M is and how you obtain it.

   (a) $|2x_1 - 5x_2| \geq 6$ (Hint: Think of this as a yes/no constraint.)
   (b) $|-x_1 + 3x_2| \leq 6$ (Hint: You won't need integer variables here.)
   (c) The variables x_1 and x_2, both are restricted to be integers, and you want to ensure that if $x1 < 2$, then $x2 \leq 5$.

(d) The variables x1 and x2, both are restricted to be integers, you want to ensure that either $x_1 + 5x_2 \leq 10$, or $3x_1 + 7x_2 \geq 12$, or both constraints are satisfied by x1 and x2.

6. Suppose you manage emergency services for a county which consists of 6 cities. You need to determine in which cities to put fire stations given that every city must be within 15 minutes of a fire station. You're given the following matrix representing the time it takes to travel between cities:

	City 1	City 2	City 3	City 4	City 5	City 6
City 1	0	10	20	30	30	20
City 2		0	25	35	20	10
City 3			0	15	30	25
City 4				0	20	15
City 5					0	30

So this matrix tells us that City 1 can be served by a fire station in cities 1 and 2. City 2 can be served by a fire station in cities 1,2 and 6. You also assume that

- fire stations don't have a capacity, and
- you can only build fire stations in cities - i.e. you can't build one in between two cities.

Because of your limited budget you want to build as few fire stations as possible while still being able to serve each city within 15 minute time. Formulate this problem as Integer Program.

7. (Logical Operators) Let x_1, x_2, $\ldots$, x_n be n binary variables and let z be some other binary variable.

- Formulate the logical operator *or* using integer linear program, i.e., write linear constraints so that (1) z takes the value of 1 if at least one of the x_i's takes the value of 1, (2) if all the x_i's take the value of zero, then z takes a value of 0.
- Formulate the logical operator *or* using integer linear program, i.e., write linear constraints so that (1) z takes the value of 1 if at least one of the x_i's takes the value of 1, (2) if all the x_i's take the value of zero, then z takes a value of 0.

8. Let x and y be binary variables. Remember:

- We say '$z = x$ or y' if z is 0 whenever both x and y are 0 and z is 1 otherwise.
- We say '$z = x$ and y' if z is 1 whenever both x and y are 1 and z is 0 otherwise.
- We say '$z =$ not x' if z is 1 when $x = 0$ and z is 0 when $x = 1$.

Let u, v, w be binary variables. Model

$$(u \text{ and } \text{not}(v)) \text{ or } w$$

using linear constraints. You may use new binary variables. (No need to give any objective function.)

9. (Warehousing Problem). Consider the following problem: There are m clients $\{1, \dots, m\}$. There are n potential locations for opening warehouses $\{1, \dots, n\}$. Up to p warehouses can be opened. The distance between the ith client and jth warehouse location is d_{ij}. Each client must be served by exactly one warehouse. Formulate a integer linear program to minimize the largest distance between a client and the warehouse serving it. Suppose now that

   - Each client need not be served. However, if a client is served, then it must be served by exactly one warehouse.
   - The revenue generated by serving client i is p_i.
   - Cost of opening warehouse j is c_j.
   - If the distance between a client and a warehouse serving it is strictly greater than r, then an additional cost t is incurred.
   - Formulate a integer linear program to decide which warehouse to open and which clients to be served so as to maximize the revenue.

10. (Dinner Host) You are asked to determine the seating arrangement of 8 diners: 4 girls (`A`, `B`, `C`, `D`) and 4 boys (`E`, `F`, `G`, `H`) on a circular table. The following conditions must be satisfied.

    - No boy must be seated next to a boy.
    - `A` and `E` should not be seated next to each other.
    - `C` and `H` should be seated next to each other.
    - If `A` and `F` are seated next to each other, then `D` and `F` must be seated next to each other.

Formulate a integer linear program to determine if a feasible seating arrangement exists.

11. (Pizza Delivery)

    - A pizza shop has to deliver pizza to n customers $\{1, ..., n\}$.
    - There is one pizza delivery car to deliver pizza to the customers.
    - t_{ij} is the time it takes to travel from customer i to customer j. t_{0i} is the time it takes to travel from pizza shop to customer i. Similarly t_{i0} is the time it takes to travel from customer i to pizza shop.
    - Since the city has a number of one way streets and the delivery is happening at peak traffic period, t_{ij} is not necessarily equal to t_{ji} (and the same for t_{i0} and t_{0i})
    - The pizza delivery car starts at the pizza shop and visits one customer after another (each customer visited exactly once). After visiting each customer it returns back to the pizza shop.

    (a) The Pizza store manager would like to have the Pizza delivered hot. Therefore, formulate a integer linear program to determine path for the delivery car so that- *each customer is visited exactly once and the last customer receiving pizza, receives it as soon as possible.*

    (b) The Pizza delivery person would like to finish the job as soon as possible which involves returning the car to the store after delivery of pizza. Therefore, formulate a integer linear program to determine path for the delivery car so that the delivery person *visits each customer exactly once and returns to the pizza shop at the earliest.*

12. Model the following problem minimizing the total cost. The model should be a mixed integer linear problem (i.e. all the constraints should be linear and some or all of the variables are allowed to be integer.)

    - A company knows its demand for the next 6 months. Let $i = 1, ..., 6$ represent months. The demand is d_i for the i^{th} month where $i \in \{1, ..., 6\}$.
    - All demands must be met. In each month, demand is met by using a combination of either production in that month or inventory

stored from previous months (or both). Therefore, in each month the company must decide whether to produce or not to produce.

- It production occurs in period i, then at least l units must be produced. (l is a fixed positive number that the production engineers have pre-specified.)
- The per unit cost of storing inventory from month i to month $i+1$ is h_i.
- The per unit cost of production in month i is c_i.
- There is a fixed cost f_i of starting up the machinery in month i. The fixed cost applies in month i only if there is non-zero production in month i and there is no production in month $i-1$. There is no fixed cost for month 1. For example, if the production in months 1, 2, 3, 4, 5, and 6 are 10, 20, 0, 5, 13, and 0 respectively, then fixed cost applies only for month 4.

Clearly define each variable.

13. A power plant has four boilers. If a given boiler is operated, it can be used to produce a quantity of steam (in tons) between the minimum and maximum given in Table 1.1. The cost of producing a ton of steam on each boiler is also given, as well as the fixed cost of operating each boiler. Steam from the boilers is used to produce power on three turbines. If operated, each turbine can process an amount of steam (in tons) between the minimum and maximum given in table 1.2. The cost of processing a ton of steam and the power produced by each turbine is also given. The power plant must produce at least 9,000 Kwh of power.

Boiler #	Min.	Max.	Cost per ton ($)	Fixed cost ($)
1	400	900	9	160
2	500	700	7	200
3	300	600	8	190
4	240	800	6	250

Table 1.1: Data for each of the boilers

- Formulate a linear IP that can be used to minimize the cost while satisfying all constraints. Clearly specify what are the decision

Turbine #	Min.	Max.	Kwh per ton of steam	Processing Cost per Ton ($)
1	100	700	7	9
2	400	800	3	4
3	300	600	6	6

Table 1.2: Data for each of the turbines

variables (and what they mean), as well as the objective function and constraints.

- Write linear constraints that correctly model the following additional restriction: If both boilers 1 and 2 operate, then boilers 3 and 4 must not operate. (you may add additional variables if you need to.)
- Suppose that if we operate both boilers 3 and 4, then instead of the fixed cost being 440, it drops to 340 since they are able to share some resources. How would you modify your linear IP formulation to reflect this cost change? (you may add additional variables if you need to.)

14. Jack and Jill love to watch DVDs and, as a couple, have vowed to always watch their DVDs together as a form of spending time with each other. They have subscribed for a 1 month free trial and estimated that they will watch 20 DVDs by the end of the free trial. They have pre-selected 100 DVDs that they would like to watch and now want to decide which 20 out of these 100 to watch during the free trial month. These 100 DVDs are divided as follows: DVDs 1-20 are action movies, DVDs 21-40 are romantic comedies, DVDs 41-60 are TV series, DVDs 61-80 are documentaries and DVDs 81-100 are science fiction.

    They associated an estimated common satisfaction index s_i to each DVD and wish to maximize their estimated common satisfaction during this month. They also established the following constraints so that each of them will not be too unhappy: They must choose at least 3 action movies and at least 3 romantic comedies, but not more than 5 science fiction and no more than 2 documentaries.

    (assume that they will be able to obtain all movies they request)

    a) Write an Integer programming formulation that Jack and Jill can use to solve this problem of maximizing their estimated common satisfaction. Clearly indicate your decision variables and what they mean.

b) Write one or more linear constraints to express the following additional conditions: (you may add decision variables if necessary, as long as you make it clear what they mean. Also, if you use a Big-M constant, explicitly state its value and how you got to it)

   i. Since DVDs 11, 12 and 13 are part of a series, either they get all of them, or they get none of them.

   ii. DVDs 41–50 are respectively seasons 1–10 of Friends, so they have decided that if they watch one season of Friends, they must watch all seasons before it. For example, if they watch season 6, they must also watch seasons 1–5.

   iii. If they watch at least 2 DVDs out of 24, 34 and 77, then they must not watch more than one DVD out of 27, 35 and 79.

c) Jack and Jill realized that their satisfaction model for DVDs 84 and 85 is slightly different. If they watch exactly one DVD among 84 and 85, then the satisfaction index is t. Else it is zero. How would you model this fact using only linear constraints and linear objective function? (you may add auxiliary variables)

15. You have been assigned to arrange the songs on the cassette version of Madonna's latest album. A cassette tape has two sides (1 and 2). The songs on each side of the cassette must total between 14 and 16 minutes in length. The length and type of each song are given in the table below:

Song	Type	Length (minutes)
1	Ballad	4
2	Hit	5
3	Ballad	3
4	Hit	2
5	Ballad	4
6	Ballad	3
7		5
8	Ballad and Hit	4

The assignment of songs to the tape must satisfy the following conditions:

(a) Each side must have exactly two ballads.

(b) Side 1 must have at least three hit songs.

(c) Either song 5 or song 6 must be on side 1.

(d) If songs 2 and 4 are on side 1, then song 5 must be on side 2.

Explain how you could use an integer programming formulation to determine whether there is an arrangement of songs satisfying these restrictions.

16. (a) There are eight machines types of A, B, C, D, E, F, G, H.

    (b) Machines A, B, C, D are used for drilling and machines E, F, G, H are used for finishing.

    (c) Each drilling machine must be coupled to exactly one finishing machine.

    (d) On the other hand each finishing machine can be coupled to 2 or less drilling machines (and can also be zero).

    (e) If drilling machine i is exclusively coupled to finishing machine k, then the cost of c_{ik}.

    (f) If on the other hand the drilling machine i and j are coupled to machine k, then the cost of c_{ijk}.

    Write a formulation that assigns each drilling machine to at least one finishing machine at minimum cost.

17. (Textbook) A company has 10 employees, each of whom can work on at most 2 team projects. Six projects are under consideration (although the company may decide to not work on each project). Each project requires 4 out of the 10 workers. The required employees for each project and revenue generated from each project is presented below. Each worker who is used on any project must be paid the retainer

Table 1.3: Required number of workers and revenue

Project	Required Employee	Revenue
1	1,4,5 8	10000
2	2,3,7, 10	15000
3	1, 6, 8, 9	6000
4	2,3,5, 10	8000
5	1,6,7, 9	12000
6	2,4,8,10	9000

shown below. Finally, each worker on a project is paid the project fee

	1	2	3	4	5	6	7	8	9	10
Retainer	800	500	600	700	800	600	400	500	400	500

	1	2	3	4	5	6	7
Fee	250	300	250	300	175	180	300

shown below.

18. At a machine tool plant, five jobs must be completed each day. There are 5 machines. The time for doing job i in machine j is t_{ij}. If a machine is used at all, there is a setup time of s_j.

    - (Textbook) The company's goal is to minimize the sum of the setup and machine operation times needed to complete all jobs. Formulate this as an IP.
    - Different machines will run for different amounts of time depending on the setup time and the jobs there are completing. The company's goal is to minimize the maximum time over the different machines. Formulate this as an IP.

19. The Transylvania Olympic Gymnastics Team consists of 6 people. Transylvania must choose 3 people to enter both the balance beam and oor excercises events. They must also enter a total of 4 people in each event. The score that gymnast j can attain in balancing is b_j and in floor excercises is f_j. Formulate an IP to maximize the total score attained by the Transylvania gymnasts.

20. Consider the following variant of lot-sizing problem.

    - The planning horizon is 6 months. Let $i = 1, ..., 6$ represent months.
    - The per unit cost of production in month i is c_i.
    - The per unit cost of storing inventory from month i to month $i + 1$ is h_i.
    - There is a fixed cost f_i of starting up the machinery in month i, i.e., if amount of production in month i is greater than zero then fixed cost is f_i and if it is zero then the fixed cost if 0.

- The company knows its demand for the next 6 months. The demand is d_i for the i^{th} month.
- There is flexibility in meeting the demand. In each month one of the following three choices must be selected:
  (a) Demand d_i is met in month i.
  (b) Demand d_i is met in next month, i.e. month $i + 1$. Essentially this choice allows the company to delay the delivery of demand by 1 month. This option is not available for the last month. Selecting this choice involves paying a penalty p_i.
  (c) Demand d_i is never met. Selecting this choice involves paying a penalty q_i.
- It must be ensured that at least demands of three months are met (either at correct time or with a delay of 1 month).

Formulate an IP to minimize cost.

`Example:` Following is an example of a valid plan for meeting demands:

(a) Demand delivered in month 1: 0 (Demand d_1 delayed by one month)
(b) Demand delivered in month 2: d_1
(c) Demand delivered in month 3: d_2 (Demand d_2 delayed by one month, Demand d_3 never met)
(d) Demand delivered in month 4: 0 (Demand d_4 is delayed by one month)
(e) Demand delivered in month 5: $d_4 + d_5$
(f) Demand delivered in month 6: 0 (Demand d_6 never met)

Other than the production cost, fixed costs, inventory costs, the penalty cost in this case is: $p_1 + p_2 + q_3 + p_4 + q_6$.

21. (Xpress Manual) Mr. Miller is in charge of establishing the weekly timetable for two sections of the last year in a college. The two sections have the same teachers, except for mathematics and sport. In the college all lessons have a duration of two hours. Furthermore, all students of the same section attend exactly the same courses. From Monday to Friday, the slots for courses are the following: $8:00$ - $10:00$, $10:15$ - $12:15$, $14:00$ - $16:00$, and $16:15$ - $18:15$. The following table lists the number of two-hour lessons that every teacher has to teach the students of the two sections per week.

Teacher	Subject	Lessons/week for section 1	Lessons/week for section 2
Ms Cheese	English	1	1
Mr Insulin	Biology	3	3
Ms map	Geography	2	2
Mr Efofecks	Mathematics	0	4
Ms Derivate	Mathematics	4	0
Mr Electron	Physics	4	3
Ms Wise	Philosophy	1	1
Mr Muscle	Sport	1	0
Ms Biceps	Sport	0	1

The sport lessons have to take place on Thursday afternoon from 14:00 to 16:00. Furthermore, the first time slot on Monday morning is reserved for supervised homework. Mr Efofecks is absent every Monday morning because he teaches some courses at another college. Mr Insulin does not work on Wednesday. And finally, to prevent students from getting bored, every section may only have one two-hour lesson per subject on a single day. Write a mathematical program that allows Mr Miller to determine the weekly timetable for the two sections.

22. A hospital is scheduling nurses for aiding in surgery. 10 surgeries need to be performed. Each surgery requires exactly 2 nurses. There are 25 available nurses. We have the following additional constraints.

    (a) A nurse can aid in a surgery only if the nurse is qualified to do so. The following data is available: $d_{ij} = 1$ if nurse i is qualified to aid in surgery j and $d_{ij} = 0$ if nurse i is not qualified to aid in surgery j (d_{ij} provided for all $i = 1, ...25$ and for all $j = 1, ..., 10$).

    (b) Nurse 1, Nurse 2, ..., Nurse 15 are senior nurses. Nurese 16, ..., Nurse 25 are junior nurses. Each surgery must have at least 1 senior qualified nurse.

    (c) Nurse 1 and Nurse 20 do not work well together. They must not be scheduled to the same surgery.

    (d) Each nurse can aid in at most 2 surgeries.

    (e) At least 5 junior nurses must be aiding surgeries.

    Formulate an integer program to find a feasible schedule.

23. Variables x and y must belong to the set drawn in Figure.

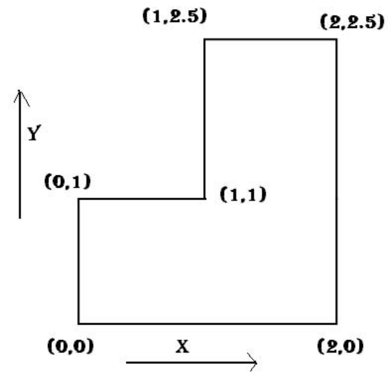


Figure 1.5:

(a) Is the allowable set of x and y convex?

(b) Write down a mixed integer programming model (by using additional variables), so that x and y belong only to the allowable set.

24. There are n supply stations and m demand centers. Let $i = 1, ..., n$ represent the supply stations and let $j = 1, ..., m$ be the demand centers. Each supply station has a capacity of at most s_i tons of products and each demand center requires at least d_j tons of products. (You can model the amount of product as a continuous variable.) The products are shipped by using trucks. Each truck can carry at most b tons of products. The cost of sending a truck from supply station i to demand center j is c_{ij}, regardless of the amount of product carried by the truck. Note that (1) Each truck travels from exactly one supply station to exactly one demand node. (2) Any number of trucks can be sent from supply station i to demand center j. Write a mixed integer linear program to minimize the total cost.

25. A company sells seven types of boxes, ranging in volume from 17 to 33 cubic feet. The demand and size of each box type are given in Table 1.4. The cost (in dollars) of producing each box is equal to the box's volume. Moreover, a fixed cost of \$1000 is incurred to produce any of a particular box type. [For example: If 50 boxes of 26 cubic feet are manufactured, then the total cost = 1000 + 26*50]. If the company desires, a demand for a box may be satisfied by a box of larger size.

	1	2	3	4	5	6	7
Size	33	30	26	24	19	18	17
Demand	400	300	500	700	200	400	200

Table 1.4: Size and Demand of the Boxes.

(a) Formulate an integer linear program to minimize the cost of meeting the demand of the boxes.

(b) Suppose the fixed cost structure is a bit different for boxes of size 19 and 17 cubic feet: If both box type 17 and 19 are manufactured, then the fixed cost is \$1500 instead of \$2000. If only one type among these two is produced, then the fixed cost remains \$1000. Write the updated and additional constraints (and variables) needed to model this.

(c) Suppose that the demands are flexible: The company must manufacture boxes to safisfy the demand of at least five out of the seven box types only. Write the updated and additional constraints (and variables) needed to model this.

26. A large-scale grocery retailer must purchase onions for two of their stores. Onions can be purchased from three farms. Here are the relevant details:

(a) Store 1 requires at least 1000 units and store 2 requires at least 2000 units of onions.

(b) Farms 1 sells at \$3 per unit and Farm 2 sells at \$4 per unit.

(c) Farm 3 sells onions in the following fashion: The first 300 units sells for \$3 per unit, the next 400 units sells at a discounted rate of \$2.5 per unit. However, the price per unit increases after the first 700 units to \$5, since farm believes that there may be more demand than supply. (Example: If 800 units are purchased from Farm 3, then the price charged by Farm 3 is $3 \times 300 + 2.5 \times 400 + 5 \times 100$.)

(d) The cost of transportation per unit from the farms to the stores are given below:

	Farm 1	Farm 2	Farm 3
Store 1	\$ 1	\$ 1	\$ 2
Store 2	\$ 2	\$ 1	\$ 1

Write a mixed integer programs to minimize total cost for the retailer.

27. The problem of finding a factorizing of an integer number N, i.e., finding integers x and y such that $N = x \times y$ with $x \geq 2$ and $y \geq 2$ is a very important problem. Formulate this problem as a linear integer program using no more than $\mathcal{O}(\log_2(N))$ variables.

28. Sudoku.
    I am sure many of you have seen “sudoku”. In a sudoku puzzle, there is a 9×9 grid. Each square in the 9×9 grid must contain a digit from 1 through 9, with the following restrictions:

    (a) Each row must contain exactly one of each digit (1-9)

    (b) Each column must contain exactly one of each digit (1-9)

    (c) Each of the nine 3×3 boxes (shown by the bold lines below) must contain exactly one of each digit (1-9).

    Some squares are already filled in (See Figure 1 on next page); your job is to fill in each of the others so that all three of the above rules are satisfied. Develop an IP model to solve sudoku.

		3	6					8
	2			4			3	
5		8		3		4		
1					7			
	4	7		9		1	8	
			2					7
		1		2		7		6
	5			6			2	
6					9	8		

Figure 1.6: sudoku

29. (J. P.-P. Richard) Paperboys.
    Two paperboys must deliver newspaper in Manhattan. They pick the papers at 6:00 am and have to deliver them to n customers as soon as possible. It is well-known that the streets in Manhattan form a rectangular grid so we can assume that the distance between two points

(x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$. The coordinates of the customers are:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	-60	40	-20	10	-30	50	-50	10	-40	20	-5	40	-50	20	15
y	40	50	-40	60	20	-30	-10	-40	10	-30	5	20	30	40	-15

The depot is located at $(0, 0)$. Assume that the paper boys cover 100 distance units per hour. Write a IP model to minimize the time for when the latest of the two paperboys returns back to the depot.

30. (Wolsey) Removing a point.
Consider the polytope P given by:

$$\begin{aligned} 2x_1 + 3x_2 &\leq 100 \\ x_1 &\geq 0 \\ x_2 &\geq 0. \end{aligned}$$

Notice that the point $(5, 10)$ is a feasible point for P. Write an integer programming formulation whose feasible set is exactly the the set of integer points in P except $(5, 10)$. (No need to give any objective function.)

31. Genetic material from different species is often "aligned" by biologists to understand the common features between them. Consider the following concrete (mathematical) version of this problem.

Two strings of characters from the set $\mathcal{C} := \{1, 2, 3, 4\}$ of lengths n_1 and n_2 are given as input. We are allowed to insert *gaps* in the strings so that the two strings become of equal length n. ($n \geq \max\{n_1, n_2\}$). Note that the strings cannot be otherwise changed (for example, the order in which the characters appear in the original strings must be the same order in which the characters appear in the new strings if the gaps are removed). This pair of new strings of equal lengths is called *an alignment*. Let's represent the gaps with the character "0" and let $\bar{\mathcal{C}} := \{0, 1, 2, 3, 4\}$. We represent the resulting first sequence (with gaps) as $a_1,\ a_2,\ a_3,\ ...,\ a_n$ and the resulting second sequence (with gaps) as $b_1,\ b_2,\ b_3,\ ...,\ b_n$, i.e. $a_j, b_j \in \bar{\mathcal{C}}\ \forall j \in \{1, \ldots, n\}$.

`Example:` String 1: 1 2 3 1 2. String 2: 2 1 4 4. We have $n_1 = 5$ and $n_2 = 4$. We may insert gaps in the two strings to obtain an alignment with $n = 7$ as follows:

$$\begin{array}{ccccccc} 1 & 0 & 0 & 2 & 3 & 1 & 2 \\ 0 & 2 & 1 & 0 & 4 & 4 & 0. \end{array} \tag{1.17}$$

Note that for a fixed value of n, many alignments are possible based on where the gaps are inserted.

We want the best possible alignment. In order to evaluate the quality of an alignment, we compare the characters at the same position. More formally, we are given a cost function $f : \bar{C} \times \bar{C} \to \mathbb{Z}_+$ and the cost of an alignment is

$$\sum_{j=1}^{n} f(a_j, b_j).$$

`Example (contd.):` Suppose f is the matrix

	0	1	2	3	4
0	100	20	30	40	50
1	20	0	50	40	10
2	30	50	0	20	70
3	40	40	20	0	30
4	50	10	70	30	0

Then the cost of the alignment given in (1.17) is $20 + 30 + 20 + 30 + 30 + 10 + 30 = 170$.

Given two sequences and a cost function matrix, write a mixed integer linear formulation to determine the alignment with minimum cost. (Assume that $f(0, 0)$ is positive and so a_j and b_j cannot be simultaneously 0 and consequently $n \leq n_1 + n_2$.)

32. You are given a connected graph $G = (V, E)$, weight $w : E \to \mathbb{R}_+$ and special subset of vertices $U \subseteq V$. The Steiner tree problem is to find a minimum-weight tree that includes all the vertices in U. Write down an IP formulation for this problem.

33. *(Alternative formulation) Let $A \in \mathbb{Z}_+^{m \times n}$ be a non-negative integral matrix and $b \in \mathbb{Z}_+^m$. Let $S := \{x \in \mathbb{Z}^n \mid Ax = b; x \geq 0\}$. Show that there exists $c \in \mathbb{Z}_+^n$ and $d \in \mathbb{Z}$ such that

$$S = \{x \in \mathbb{Z}^n \mid c^T x = d, x \geq 0\}.$$

Hint: Consider aggregating rows with suitable integer multipliers so that the set of integer feasible solutions does not change.

34. (Comparing formulations) Problem 17, Page 25, Textbook.

35. Write an integer programming formulation for the following problem: Jobs $\{1, \ldots, n\}$ must be processed on a single machine. Each job is available for processing after a certain time, called release time. For each job we are given its release time r_i, its processing time p_i and its weight w_i. Formulate as an integer linear program the problem of sequencing the jobs without overlap or interruption so that the sum of the weighted completion times is minimized.

36. Office design problem: We are given a rectangular room of length along x-axis L and along y-axis W. We want to place n cubicles inside this room. You must design the center (c_i^x, c_i^y) and the length along x-axis l_i and length along y-axis w_i of the i-th cubicle. Following constraints must be satisfied:

    - Size of cubicle: The sum of l_i and w_i must be at least a_i.
    - Aspect ratio of cubicle: $\min\{l_i/w_i, w_i/l_i\} \geq 1/2$.
    - No two cubicles can overlap.
    - All the cubicles must fit inside the rectangular room.

    The objective in minimize the sum of the pairwise Manhattan distance between centers i.e., $\sum_{1 \leq i < j \leq n} |c_i^x - c_j^x| + |c_i^y - c_j^y|$. Write a mixed integer linear model for the above office design problem.

37. Let $S := \{x \in \{0,1\}^4 \,|\, 90x_1 + 35x_2 + 26x_3 + 25x_4 \leq 138\}$. Show that $S = \{x \in \{0,1\}^n \,|\, 2x_1 + x_2 + x_3 + x_4 \leq 3\}$ and $S = \{x \in \{0,1\}^4 \,|\, 2x_1 + x_2 + x_3 + x_4 \leq 3, x_1 + x_2 + x_3 \leq 2, x_1 + x_2 + x_4 \leq 2, x_1 + x_3 + x_4 \leq 2\}$. Can you rank these formulations in terms of the tightness of their linear relaxation. Show any strict inclusion.

38. (Formulation; comparison of formulation) Consider a piecewise linear function f defined by the following break points: $(a_i, f(a_i))$ $i \in$

$\{1, \ldots, N\}$. Consider the following model:

$$
\begin{aligned}
x &= a_1 + \sum_{i=1}^{N-1} u_i && (1.18)\\
f(x) &= f(a_1) + \sum_{i=1}^{N-1} \frac{f(a_{i+1}) - f(a_i)}{a_{i+1} - a_i} u_i && (1.19)\\
u_1 &\leq a_2 - a_1 && (1.20)\\
u_{N-1} &\geq 0 && (1.21)\\
u_i &\leq (a_{i+1} - a_i) v_{i-1} \ \forall i \in \{2, \ldots, N-1\} && (1.22)\\
u_i &\geq (a_{i+1} - a_i) v_i \ \forall i \in \{1, \ldots, N-2\} && (1.23)\\
1 &\geq v_1 \geq v_2 \geq \cdots \geq v_{N-2} \geq 0 && (1.24)\\
v_i &\in \{0, 1\} \ \forall i \in \{1, \ldots, N-2\} && (1.25)
\end{aligned}
$$

(a) Prove that the above MIP model is also a correct formulation of piecewise linear problem.

(b) Prove that the MIP model discussed in class is a better model that the above by comparing the LP relaxation value in the space of x and $f(x)$.

Chapter 2

Introduction to Computational Complexity

2.1 Standard encoding

We begin by discussing the binary encoding of rational numbers.

Definition 2.1 (Rational number). *A* ***rational number*** α *is a number that can be written as* $\alpha = \frac{p}{q}$*, where* $p \in \mathbb{Z}$*,* $q \in \mathbb{N}$ *and* $\gcd(p,q) = 1$*. We denote the set of all rational numbers by* $\mathbb{Q}$*.*

Examples of rational numbers are $2, \frac{-1}{7}, \frac{6}{5}, -49$. On the other hand, the numbers $\sqrt{2}, \pi$ and $\ln 7$ are examples of numbers that are *not* rational (i.e., irrational).

Rational numbers are extremely important in Integer Programming Theory, as we have seen in home work 1. For now, we use the definition of rational numbers to define what we call the *encoding size* of a rational number, which essentially quantifies the number of bits required for storing a rational number in binary representation.

Consider first an integer number $p \in \mathbb{Z}$. The **encoding size** of p (denoted by $\mathrm{size}(p)$) is computed as

$$\mathrm{size}(p) = 1 + \lceil \log_2(|p| + 1) \rceil \tag{2.1}$$

where 1 bit is for storing the sign of p, and $\lceil \log_2(|p| + 1) \rceil$ bits to store $|p|$ in base 2. Extending this idea, in order to store $\alpha = \frac{p}{q} \in \mathbb{Q}$ we need to store the sign of p, and $|p|$ and q in base 2 (recall that $q \in \mathbb{N}$ and thus we do not need to store its sign). It follows that

$$\text{size}(\alpha) = \text{size}\left(\frac{p}{q}\right) = \text{size}(p) + \text{size}(q) \tag{2.2}$$

$$\overset{(2.1)}{=} 1 + \lceil \log_2(|p| + 1) \rceil + \lceil \log_2(q + 1) \rceil. \tag{2.3}$$

We can further extend this definition for storing vectors and matrices of rational entries. For $a = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} \in \mathbb{Q}^n$ with $a_i = \frac{p_i}{q_i}, p_i \in \mathbb{Z}, q_i \in \mathbb{N}, \forall i \in [n]$, we define

$$\text{size}(a) = \sum_{i=1}^{n} \text{size}(a_i) = \sum_{i=1}^{n} \text{size}\left(\frac{p_i}{q_i}\right) \tag{2.4}$$

$$\overset{(2.3)}{=} \sum_{i=1}^{n} \left(1 + \lceil \log_2(|p_i| + 1) \rceil + \lceil \log_2(q_i + 1) \rceil\right)$$

$$= n + \sum_{i=1}^{n} \left(\lceil \log_2(|p_i| + 1) \rceil + \lceil \log_2(q_i + 1) \rceil\right). \tag{2.5}$$

Similarly, for $A \in \mathbb{Q}^{m \times n}$ with $A_{ij} = \frac{p_{ij}}{q_{ij}}, p_{ij} \in \mathbb{Z}, q_{ij} \in \mathbb{N}, \forall i \in [m], \forall j \in [n]$, we define the encoding size of a rational matrix A as

$$\text{size}(A) = \sum_{i=1}^{m} \sum_{j=1}^{n} \text{size}(A_{ij}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \text{size}\left(\frac{p_{ij}}{q_{ij}}\right) \tag{2.6}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} \text{size}(p_{ij}) + \text{size}(q_{ij})$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} \left(1 + \lceil \log_2(|p_{ij}| + 1) \rceil + \lceil \log_2(q_{ij} + 1) \rceil\right)$$

$$= mn + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\lceil \log_2(|p_{ij}| + 1) \rceil + \lceil \log_2(q_{ij} + 1) \rceil\right). \tag{2.7}$$

As an example, consider the graph in Figure 2.1. Recall that a graph $G = (V, E)$ can be represented by its underlying *adjacency matrix* $A \in \{0, 1\}^{|V \times V|}$, where $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise. In the given example we can represent the graph by the adjacency matrix

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

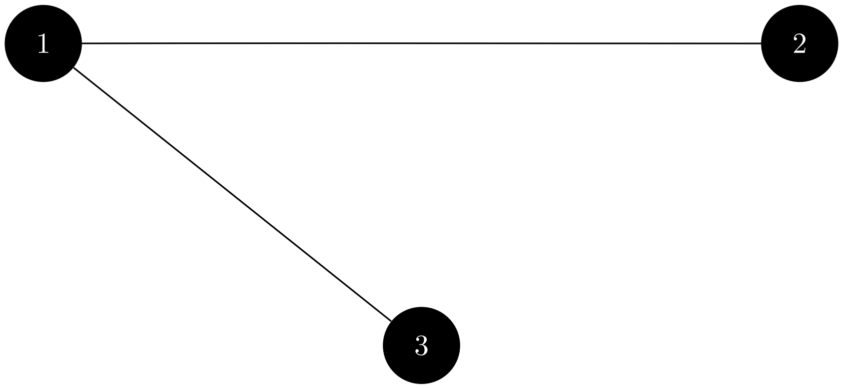


Figure 2.1: A simple graph

We can conclude that the encoding size of a graph $G = (V, E)$ is in fact the encoding size of its adjacency matrix.

Lemma 2.1. *Let $A \in \mathbb{Q}^{n \times n}$ such that size$(A) = \theta$. Then*

$$\textit{size}(\det(A)) \leq 2\theta$$

Proof. As $A \in \mathbb{Q}^{n \times n}$, it is clear that $\det(A) \in \mathbb{Q}$ and therefore we can find $p \in \mathbb{Z}, q \in \mathbb{N}$ such that $\det(A) = \frac{p}{q}$. Similarly, for each $(i, j) \in [n] \times [n]$ we can find $p_{ij} \in \mathbb{Z}$ and $q_{ij} \in \mathbb{N}$ such that $A_{ij} = \frac{p_{ij}}{q_{ij}}$. Note that we want to prove

$$\text{size}(\det(A)) = 1 + \lceil \log_2(|p| + 1) \rceil + \lceil \log_2(q + 1) \rceil \leq 2\theta$$

Note that by definition of θ we have

$$\begin{aligned}\theta &\overset{(2.7)}{=} n^2 + \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lceil \log_2(|p_{ij}|+1)\rceil + \lceil \log_2(q_{ij}+1)\rceil\right)\\ &\overset{n>1}{>} 1 + \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lceil \log_2(|p_{ij}|+1)\rceil + \lceil \log_2(q_{ij}+1)\rceil\right)\end{aligned}$$

$$\begin{aligned}\implies \theta - 1 &> \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\lceil \log_2(|p_{ij}|+1)\rceil + \lceil \log_2(q_{ij}+1)\rceil\right) \qquad (2.8)\\ &> \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\log_2(|p_{ij}|+1) + \log_2 q_{ij}\right)\end{aligned}$$

$$\overset{2^{(\cdot)}}{\implies} 2^{\theta-1} > 2^{\sum_{i=1}^{n}\sum_{j=1}^{n}(\log_2(|p_{ij}|+1)+\log_2 q_{ij})} = \prod_{i=1}^{n}\prod_{j=1}^{n} q_{ij}(|p_{ij}|+1) \tag{2.9}$$

and therefore

$$2^{\theta-1} > \prod_{i=1}^{n}\prod_{j=1}^{n} q_{ij} \tag{2.10}$$

Now we prove an auxiliary claim.

Claim 2.1.

$$q \leq \prod_{i=1}^{n}\prod_{j=1}^{n} q_{ij} \tag{2.11}$$

Proof. Using *Leibniz Formula* we can write $\det(A)$ as

$$\det(A) = \frac{p}{q} = \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma)\prod_{i=1}^{n} A_{\sigma(i),i} = \frac{\bar{p}}{\prod_{i=1}^{n}\prod_{j=1}^{n} q_{ij}} \tag{2.12}$$

where the denominator on the RHS of (2.12) is a common multiple of the denominators of all the rational entries of A, obtained from all the additions in *Leibniz Formula.* Certainly since $\gcd(p, q) = 1$ (by definition of the rational number $\det(A)$), then q is the least denominator that can be used to write $\det(A)$ as a fraction, following that $q \leq \prod_{i=1}^{n}\prod_{j=1}^{n} q_{ij}$, as desired. $\square$

Combining (2.10) and (2.11) we conclude that

$$q < 2^{\theta-1}. \tag{2.13}$$

As $q \in \mathbb{N}$ and $\theta \in \mathbb{N}$ (and therefore $2^{\theta-1} \in \mathbb{N}$), it follows from (2.13) that

$$\begin{aligned} q < 2^{\theta-1} \implies q+1 \leq 2^{\theta-1} &\implies \log_2(q+1) \leq \theta - 1 \\ &\overset{\theta-1\in\mathbb{Z}}{\implies} \lceil \log_2(q+1) \rceil \leq \theta - 1. \end{aligned} \tag{2.14}$$

Now note that

$$\begin{aligned} \det(A) = \frac{p}{q} \implies |p| = q|\det(A)| &\overset{(*)}{\leq} q \prod_{i=1}^{n} \sqrt{\sum_{j=1}^{n} A_{ij}^2} = q \prod_{i=1}^{n} \sqrt{\sum_{j=1}^{n} \left(\frac{p_{ij}}{q_{ij}}\right)^2} \\ &\leq q \prod_{i=1}^{n} \sqrt{\sum_{j=1}^{n} p_{ij}^2} \\ &\leq q \prod_{i=1}^{n} \sqrt{\prod_{j=1}^{n} (|p_{ij}|+1)^2} \\ &\leq q \prod_{i=1}^{n} \prod_{j=1}^{n} (|p_{ij}|+1) \\ &\overset{(2.11)}{\leq} \prod_{i=1}^{n} \prod_{j=1}^{n} q_{ij} \prod_{i=1}^{n} \prod_{j=1}^{n} (|p_{ij}|+1) \\ &= \prod_{i=1}^{n} \prod_{j=1}^{n} q_{ij}(|p_{ij}|+1) \\ &\overset{(2.9)}{<} 2^{\theta-1}. \end{aligned} \tag{2.15}$$

where (*) follows from *Hadamard's Inequality.* We conclude from (2.15) that

$$|p| < 2^{\theta-1}. \tag{2.16}$$

As $|p| \in \mathbb{Z}$ and $\theta \in \mathbb{N}$ (and thus $2^{\theta-1} \in \mathbb{N}$), it follows from (2.16) that

$$\begin{aligned} |p| < 2^{\theta-1} \implies |p|+1 \leq 2^{\theta-1} &\implies \log_2(|p|+1) \leq \theta - 1 \\ &\overset{\theta-1\in\mathbb{Z}}{\implies} \lceil \log_2(|p|+1) \rceil \leq \theta - 1 \\ &\implies 1 + \lceil \log_2(|p|+1) \rceil \leq \theta. \end{aligned} \tag{2.17}$$

Lastly, by definition of encoding size of $\det(A)$ and by (2.14) and (2.17) we have

$$\begin{aligned}\text{size}(\det(A)) &= 1 + \lceil \log_2(|p| + 1)\rceil + \lceil \log_2(q + 1)\rceil \\ &\leq \theta + (\theta - 1) \\ &= 2\theta - 1 \\ &< 2\theta\end{aligned}$$

which is the desired result. □

For a given matrix $A \in \mathbb{Q}^{n\times n}$, this result reveals that although the value of $\det(A)$ can be very large, its encoding size is no more than twice the encoding size of A.

2.2 Problem, instance, algorithm, worst-case running time

Definition 2.2 (Problem and Instances)**.** *A **computational problem** can be viewed as an abstract question to be solved. In contrast, an **instance** of a problem is a concrete realization of such, which can be used as the input for a decision problem.*

Observation 2.1. *In the same line of the previous definition, a particular input string of a computational problem is referred to as a problem instance, and should not be confused with the problem itself.*

To illustrate the difference between the concepts of *problem* and *instance*, consider the following instance of the decision version of the Traveling Salesman Problem (TSP): Is there a route of at most 2000 kilometers that visits the 15 largest cities of Germany exactly once? The quantitative answer to this particular problem instance (characterized by the length of the asked tour, namely 2000 km; and by the cities in question, in this case the 15 largest cities of Germany) is of little use for solving other instances of the problem, such as asking for a round trip through all sites in Milan whose total length is at most 10 km and visits every site exactly once. The previous situations are two *instances* of the following *computational problem*: Given n points, can we find a tour of length at most l such that all the points are visited exactly once?

Definition 2.3 (Decision problem)**.** *A **decision problem** is a special type of computational problem whose answer is either yes or no (or equivalently, either 1 or 0).*

Observation 2.2. *A decision problem can be viewed as a formal language, where the elements of the language are instances whose output is yes, and the elements not in the language are those instances whose output is no. The objective is to decide, with the aid of an algorithm, whether a given input string is a member of the formal language under consideration. If the algorithm deciding this problem returns the answer yes, the algorithm is said to accept the input string, otherwise it is said to reject the input.*

Optimization problems such as

$$\min \; c^T x \tag{2.18a}$$
$$s.t. \; x \in P \tag{2.18b}$$

can be seen as decision problems. As an example, suppose that for a given set of cities, the set P is the polyhedron containing all the possible tours that visit every such city exactly once, and $c^T x$ represents the length of tour x. We may want to determine whether there exists a tour such that its length is at most a certain amount c_0, i.e., if there exists a tour $x^* \in P$ such that $c^T x^* \leq c_0$. We now can transform problem (2.18) into a *feasibility problem* by modifying it into the optimization problem

$$\min \; c^T x \tag{2.19a}$$
$$s.t. \; x \in P \tag{2.19b}$$
$$c^T x \leq c_0 \tag{2.19c}$$

We can then use some algorithm (e.g., Binary Search) to solve (2.19). Note that (2.19) directly solves the decision problem of finding a tour of length at most c_0: if (2.19) admits one solution, then the answer to the decision problem is "yes", whereas if (2.19) is not feasible then the answer is "no".

As we have previously referred to the term "algorithm" multiple times, we now give a somehow informal yet accepted definition.

Definition 2.4 (Algorithm)**.** *An **algorithm** is a step-by-step procedure for calculations. It is a method that consists of a finite number of exact, finite instructions. When applied to a problem of its class, it always finishes in a finite number of steps and always produces a correct answer.*

Observation 2.3. *An algorithm is expressed as a finite list of well-defined instructions for calculating a function. Starting from an initial state and initial input, the instructions describe a computation that, when executed, proceeds through a finite number of well-defined successive states, eventually producing an output and terminating at a final ending state.*

A fair question to be asked is the following: *"What makes some problems computationally hard and others easy?"* Surprisingly, although this question has been intensively researched over the last 45 years, the answer is still to be found.

As an example, consider the problems of sorting a set of numbers. This is indeed an easy problem that even small computers can solve very fast for a set of 1,000,000 numbers. On the other hand, if we consider a scheduling problem with non-overlapping constraints, even a supercomputer may require centuries to find the best schedule of an instance of only 1,000 activities.

In general, the number of steps an algorithm uses on a particular instance may depend on several parameters (e.g., if the input is a graph, the number of steps may depend on the number of nodes, the number of edges and the maximum degree of the graph). For simplicity, we compute the **running time** (also called **time complexity**) of the algorithm purely as a function of the size[1] of the instance and do not consider any other parameter, and we observe that typically a larger instance requires a larger running time for being solved[2].

Using some notation, if the input size is n, the running time of an algorithm used to solve that instance is expressed as a function of n. To do this, we assume that certain operations are executed in unit time, and we can therefore obtain the function f. For instance, if we apply **Binary Search** to sort an unsorted list of numbers which has n elements and we assume that each lookup of an element in the list can be done in unit time, then at most $log_2(n+1)$ time units are needed to return the list sorted.

We can express the running time of an algorithm A on a problem class π and on an instance of size n, as follows:

$$f(n) = \max_{\rho \in \pi, \text{size}(\rho)=n} \{\text{running time of } A \text{ on instance } \rho\}$$

There is a vast literature on Complexity Theory, and the study of algorithms can be done by employing many different efficiency measures. We limit ourselves to the most traditional one, which we define next.

Definition 2.5 (Worst-case time complexity)**.** *We define the* ***worst-case time complexity*** $T(n)$ *as the maximum time taken over all inputs of size* n.

[1] Generally taken to be the size of the input in bits.

[2] One could also consider the **space complexity** of an algorithm, namely the size of the storage location that it uses while executing its instructions

This definition is motivated by the fact that the time taken on different inputs of the same size can be different, thus we consider the worst of all such cases with such size. Other measures include **average-case analysis** and **best-case analysis**.

Because the exact running time of an algorithm often is a complex expression, it is usually estimated. For doing so we typically perform an **asymptotic analysis**, which analyzes the running time of an algorithm when run on large inputs. The usual way of doing this is by considering only the highest order term of the expression for the running time of the algorithm, disregarding both its coefficient and all the lower order terms, as the highest order term dominates the other terms on large inputs.

Consider for instance the function $f(n) = 2n^3 + 200n^2 + 4500n + 540000$. We note that f has four terms and the highest order term is $2n^3$. Disregarding its coefficient 6 we say that f is *asymptotically at most* n^3. The **asymptotic notation** or **big-O notation** for describing this relationship is $f(n) = O(n^3)$. We formalize this notion in the following definition.

Definition 2.6 (Big-O notation)**.** *Let $f, g : \mathbb{N} \to \mathbb{R}^+$. We say that $f(n) = O(g(n))$ if $\exists c, n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,*

$$f(n) \leq cg(n)$$

*When $f(n) = O(g(n))$, we say that $g(n)$ is an **upper bound** for $f(n)$, or more precisely, that $g(n)$ is an **asymptotic upper bound** for $f(n)$, to emphasize that we are suppressing constant factors.*

2.3 Theory of $\mathcal{NP}$-completeness

In this section we further study some important concepts on the Theory of Complexity, which will give us some insights on how difficult a problem may be to solve.

2.3.1 Introduction

We start with the following definition.

Definition 2.7 (Polynomial-time solvable)**.** *A problem π is said to be **polynomial-time solvable** whenever there exists an algorithm that solves π and whose running-time is $O(n^k)$, where $k > 0$ and n is the input size (bits). In words,* *the running time of the algorithm that solves π is asymptotically bounded above by a polynomial function of the input size. In such case we say that the algorithm is a **polynomial-time algorithm**.*

Remark 2.1. *If π is a polynomial-time solvable problem, we say that $\pi \in \mathcal{P}$, where $\mathcal{P}$ is known as a fundamental complexity class of problems. In particular, $\mathcal{P}$ is the class of all the polynomimal-time solvable problems.*

Example 2.1. *With the intention of facilitating the understanding of the concepts defined above, consider the following examples:*

- *The function $f(n) = n^2 \log(n)$ is bounded above by the polynomial function n^3.*
- *Examples of polynomial-time solvable problems are Sorting, Linear Programming, Totally Unimodular Integer Programs and the Lot-Size Problem.*

One may think if it is worth bothering about polynomial-time algorithms. It turns out that even if an algorithm is $O(n^5)$ or $O(n^{10})$ (which are in principle very large orders of magnitude), most of polynomial-time algorithms work well in practice, as the notation $O(n^k), k > 0$ only measures the worst-case behavior. Also, classification of problems based on this criterion is easy to understand, convenient and reasonable, reason why it is widely used in the literature. Moreover, once it is proven that a problem is polynomial-time solvable with a large exponent (e.g., $O(n^{10})$), it is common to see more researchers attempting to lower such exponent.

When confronted by a given problem, usually is not trivial for researchers to give an efficient algorithm (e.g., polynomial-time) to solve it, nor is to prove such efficient algorithm does not exist. To give a plausible answer, we could say that a problem is at least as difficult to solve as every other problem belonging to a large class of problems that are widely known as "difficult". The Theory of $\mathcal{NP}$-completeness provides us with a systematic way to illustrate such idea. Next we give a brief description.

2.3.2 $\mathcal{NP}$ and co-$\mathcal{NP}$

Definition 2.8 (Nondeterministic-polynomial time ($\mathcal{NP}$))**.** *Given a decision problem π. We say $\pi \in \mathcal{NP}$ if for every instance of π whose answer is "yes", we can point out a certificate (of the fact that the answer is "yes")* *which can be verified in polynomial time. The class $\mathcal{NP}$ is the set of all such problems.*

Example 2.2. *Consider the following examples:*

- *(TSP) As we previously discussed, the decision problem related to TSP is as follows: Given a graph $G = (V, E)$ with $|V| = n$, is there a tour x that visits all the nodes exactly once whose length is no more than a given number c_0?. If the answer is yes, then a certificate of this would be such tour x, which we could verify that visits each node exactly once (namely that it is connected and every node has degree 2), compute its distance and compare it to the given bound c_0 in polynomial time.*

- *(0-1 IP) Given $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m$ and $c \in \mathbb{Q}^n$, consider the 0-1 integer program*

$$\begin{aligned} \min \ & c^T x \\ s.t. \ & Ax \leq b \\ & x \in \{0,1\}^n \end{aligned}$$

  and the decision problem defined by the question "can we find $x \in \{0,1\}^n$ such that $Ax \leq b$ and $c^T x \leq c_0$?". Indeed if such x exists, x itself would be a certificate and we could verify if it satisfies the m constraints given by $Ax \leq b$ plus the requirement $c^T x \leq c_0$ in polynomial time (as each constraints have n linear terms, thus we could verify if x satisfies all the conditions in $O(mn)$). We will later show that general IP is also in $\mathcal{NP}$.

From the previous definition we can conclude that the certificate of a "yes" answer to a problem of the $\mathcal{NP}$ class is "small" in size, i.e., its size is bounded by a polynomial function of the input size. If the certificate was not of polynomial size, we would not be able to verify the "yes" answer in polynomial time.

In contrast (and this is crucial to understand), the process of coming up with the answer of the instance of the decision problem (i.e., *how* to determine if the answer is yes or no) is irrelevant for this definition (an example of this is the decision version of the TSP: given a number c_0, *how* to find a certificate - a tour of length at most c_0 that visits all the cities exactly once -, or prove that such certificate does not exist, is very difficult; however, *if* such tour exists and we are provided with such, the process of *verifying* that such tour satisfies the conditions can be done in polynomial time and thus the TSP is $\mathcal{NP}$). As important as this last observation, an $\mathcal{NP}$ problem is not required to have a polynomial size (polynomial-time verifiable) certificate for instances of decision problems whose answer is "no". Problems like such define another class, which we introduce next.

Definition 2.9 (Complement of $\mathcal{NP}$ problems (co-$\mathcal{NP}$)). *We say that a (decision) problem is co-$\mathcal{NP}$ if for any instance of such problem, whenever the answer is "no", then we can find a certificate of this fact that can be checked in polynomial time.*

Having defined the three classes of problems $\mathcal{P}$, $\mathcal{NP}$ and co-$\mathcal{NP}$, we present the following relationship between them.

Proposition 2.1. $\mathcal{P} \subseteq \mathcal{NP} \cap co\text{-}\mathcal{NP}$

Proof. Consider a decision problem $\pi \in \mathcal{P}$, i.e., π is polynomial-time solvable and therefore there exists an algorithm A that solves any instance of π in polynomial time.

First we prove that $\pi \in \mathcal{NP}$. Indeed, if we are given an instance ρ_1 of π whose answer is "yes", note that A itself is a certificate that can be verified in polynomial time: we can apply A on ρ_1 and since the answer to ρ_1 is "yes", A will come up with the "yes" answer in polynomial time, which will be always correct since A solves correctly any instance of π, in particular ρ_1. Therefore, $\pi \in \mathcal{NP}$.

The proof of $\pi \in$ co-$\mathcal{NP}$ is done by the same argument but applying A on an instance ρ_2 of π whose answer is "no". Therefore $\pi \in$ co-$\mathcal{NP}$.

Combining above results we conclude that $\pi \in \mathcal{P} \implies \pi \in \mathcal{NP} \cap$co-$\mathcal{NP}$, and the proposition follows. $\square$

Before ending this section, we highlight the fact that the statement $\mathcal{P} = \mathcal{NP} \cap$ co-$\mathcal{NP}$ has not been proved nor disproved, thus even after more than 40 years of complexity theory it remains to be an open question. Although most of the related research community believes that this is not true, this question is one of the most important open questions in mathematics and theoretical computer science. It is such the importance that the Clay Mathematics Institute stated this question as one of the Millennium Prize Problems, and whoever manages to prove or disprove this fact will be awarded a US $1,000,000 by this institution. In addition, you get to skip the final exam of this class.

2.3.3 $\mathcal{NP}$-Completeness

Now we will study a class of problems that are known to be "difficult" to solve. We start by an important definition.

Definition 2.10 (Polynomial-time reduction)**.** *Let π^1, π^2 be decision problems. We say π^1* ***polynomially reduces to*** *π^2, denoted as $\pi^1 \preceq \pi^2$, if there exists a mapping $\phi : \pi^1 \to \pi^2$ that maps any instance ρ of π^1 into an instance $\phi(\rho)$ of π^2 satisfying*

- *ρ is "yes" $\iff$ $\phi(\rho)$ is "yes".*
- *$(\phi(\rho))$ can be constructed in polynomial-time from (ρ).*

The mapping ϕ is called the ***polynomial-time reduction*** *of π^1 to π^2.*

Remark 2.2. *This is equivalent to the statement "Problem π^2 is at least as difficult to solve than problem π^1", i.e., if we are able to solve π^2 then we can solve π^1.*

Note that this implies the following result.

Theorem 2.1. *If $\pi^1 \preceq \pi^2$ and $\pi^2 \in \mathcal{P}$, then $\pi^1 \in \mathcal{P}$.*

Proof. Indeed, if $\pi^2 \in \mathcal{P}$, then there exists a polynomial-time algorithm A deciding π^2. Since $\pi^1 \preceq \pi^2$, there exists a polynomial time reduction ϕ from π^1 to π^2. Now let $\rho \in \pi^1$ be an arbitrary instance of π^1. The following algorithm $\bar{A}$ solves ρ in polynomial time: $\bar{A} =$ "On input ρ:
(i) Compute $\phi(\rho)$
(ii) Run A on input $\phi(\rho)$ and output whatever A outputs".

Note that $\bar{A}$ runs in polynomial time: step (i) runs in polynomial time since ϕ is a polynomial-time reduction, and step (ii) runs in polynomial time since A solves any instance of π^2 (in particular $\phi(\rho) \in \pi^2$) in polynomial time. Then the sum of the running times of steps (i) and (ii) is indeed polynomial, and thus so is the running time of $\bar{A}$. As ρ is "yes" $\iff$ $\phi(\rho)$ is "yes" (again, since ϕ is a polynomial time reduction from π^1 to π^2), it follows that whatever A decides for $\phi(\rho) \in \pi^2$ in step (ii) of $\bar{A}$ also solves the instance $\rho \in \pi^1$. As this is for an arbitrary instance $\rho \in \pi^1$ done by the polynomial time algorithm $\bar{A}$, it follows that all the instances of π^1 are polynomial-time solvable, and therefore $\pi^1 \in \mathcal{P}$. $\square$

Now we have the tools to define the classes of problems $\mathcal{NP}$-complete and $\mathcal{NP}$-hard.

Definition 2.11 ($\mathcal{NP}$-complete and $\mathcal{NP}$-hard)**.** *A problem π is said to be $\mathcal{NP}$-hard if $\tilde{\pi} \preceq \pi$, $\forall \tilde{\pi} \in \mathcal{NP}$ (which reads "problem π is at least as difficult as any problem in $\mathcal{NP}$"). If we also have $\pi \in \mathcal{NP}$, then we say that π is $\mathcal{NP}$-complete (denoted as $\mathcal{NPC}$).*

Intuitively, this definition states that any $\mathcal{NP}$-hard problem is at least as difficult as any problem in $\mathcal{NP}$. Similarly, any problem which is $\mathcal{NP}$-complete is at least as difficult as all the other $\mathcal{NP}$ problems.

- On the theoretical side, a researcher wanting to prove that $\mathcal{P} \neq \mathcal{NP}$ may focus on an $\mathcal{NPC}$ problem. Indeed, if any problem in $\mathcal{NP}$ requires more than polynomial time to be solved, so do all the $\mathcal{NPC}$ problems (since these are the most difficult problems in the $\mathcal{NP}$ class). Furthermore, a researcher attempting to prove that $\mathcal{P} = \mathcal{NP}$ only needs to find a polynomial time algorithm for solving any $\mathcal{NPC}$ problem to achieve this goal.

- Proving that a problem is $\mathcal{NP}$-complete may be "used to prevent wasting time" searching for a polynomial time algorithm to solve a particular problem.

Now we have formally defined $\mathcal{NPC}$ as the class of problems known as difficult to solve. Polynomial- time reduction provides us a tool to prove that a new problem π is $\mathcal{NPC}$ if we know at least one problem $\pi_0 \in \mathcal{NPC}$. More precisely, if we want to prove that a problem π is $\mathcal{NPC}$ and we already know a problem $\pi_0 \in \mathcal{NPC}$ then we need to prove

1. $\pi \in \mathcal{NP}$.

2. $\pi_0 \preceq \pi$.

Definition 2.12 (Satisfiability problem (SAT))**.** *Let U be the ground set of boolean variables $u_1, u_2, \ldots, u_n$. A truth assignment for U is a function $t : U \to \{T, F\}$, where T and F states for "True" and "False". If $u \in U$, then u and $\bar{u}$ (complement of u) are called **literals** over U. A **clause** over U is a set of literals over U, which represents the logical disjunction of these literals. We say that a clause is satisfied by a truth assignment if and only if at least one of the literals in it is true. Let $C = \{c_1, c_2, \ldots, c_m\}$ be a set of clauses over U. We say C is satisfiable if and only if there exists a truth assignment which satisfies all the clauses in C simultaneously.*

The SAT statement is as follows: given a set of boolean variables $U = \{u_1, \ldots, u_n\}$ *and a collection of clauses* $C = \{c_1, \ldots, c_m\}$ *over* U*. Is there a truth assignment* t *that satisfies all the clauses in* C *simultaneously?.*

Example 2.3. *Let* $U = \{u_1, u_2, u_3, u_4\}, C = \{\{u_1, \bar{u}_2\}, \{u_3\}, \{\bar{u}_1\}, \{u_2, u_4\}\}$. *The SAT consists of finding a truth assignment* $t : U \to \{T, F\}$ *such that the logic statement*

$$(u_1 \textit{ or not } u_2) \textit{ and } (u_3) \textit{ and } (\textit{not } u_1) \textit{ and } (u_2 \textit{ or } u_4)$$

is true. We can see by inspection that the answer to this question is true. Indeed, if we define t *such that*

$$\begin{aligned} t(u_1) &= F \\ t(u_2) &= F \\ t(u_3) &= T \\ t(u_4) &= T \end{aligned}$$

then we observe that

$$\begin{aligned} & (t(u_1) \textit{ or not } t(u_2)) \textit{ and } (t(u_3)) \textit{ and } (\textit{not } t(u_1)) \textit{ and } (t(u_2) \textit{ or } t(u_4)) \\ \Longleftrightarrow & (F \textit{ or not } F) \textit{ and } (T) \textit{ and } (\textit{not } F) \textit{ and } (F \textit{ or } T) \\ \Longleftrightarrow & (F \textit{ or } T) \textit{ and } (T) \textit{ and } (T) \textit{ and } (F \textit{ or } T) \\ \Longleftrightarrow & (T) \textit{ and } (T) \textit{ and } (T) \textit{ and } (T) \\ \Longleftrightarrow & T \end{aligned}$$

and therefore the requested t *exists.*

Now we see the answer to the question "where is the original $\mathcal{NPC}$ problem π_0", which is provided by the following theorem by Stephen Cook and Leonid Levin.

Theorem 2.2 (Cook)**.** *SAT is* $\mathcal{NPC}$

It turns out that the aforementioned π_0 is the SAT problem. Having knowledge of this fact, we are now capable of proving that other problems π are $\mathcal{NPC}$. We exemplify the procedure for proving so in the proof of the following theorem.

2.3.4 Binary integer program

Theorem 2.3. *Binary Integer Programming is $\mathcal{NPC}$*

Proof. We first prove that binary IP is $\mathcal{NP}$. Take an instance of a decision problem related to a binary IP. If its answer is "yes" and we are provided with a certificate of such answer (a binary vector of size n), we can evaluate it in $O(mn)$ time, since we evaluate m constraints, each with at most n terms, and then we compare each of the m resulting row with the corresponding right hand side (i.e., for a problem $Ax \leq b, x \in \{0,1\}^n$ and a certificate $\hat{x} \in \{0,1\}^n$, we can compute $A\hat{x}$ in $O(mn)$ unit operations and then compare each $a_i^T\hat{x}$ with the corresponding b_i for $i = 1, \dots, m$, this is in $O(m)$ operations). Therefore, as we can evaluate a certificate of a "yes" answer in an amount time bounded above by a polynomial function of m and n (size of the instance), it follows that binary IP is $\mathcal{NP}$. Next, we reduce SAT to a binary IP. For this we define an arbitrary instance of the problem known to be $\mathcal{NPC}$, in this case SAT[3]. Let this instance be defined by the set of literals $U = \{u_1, \dots, u_n\}$ and clauses $C = \{c_1, \dots, c_m\}$. We first need to construct a particular instance of binary IP related to the given arbitrary SAT instance[4], such that the size of this particular instance is a polynomial function of the size of the SAT instance and the binary IP is feasible if and only if the arbitrary SAT instance is satisfiable. We do this as follows:

- For each literal $u_i \in U$, define a binary decision variable $x_i \in \{0,1\}$.
- For each clause $c_j \in C$, $c_j = \{u_{j_1}, \dots, u_{j_k}, \bar{u}_{j_k+1}, \dots, \bar{u}_{j_k+l}\}$, define the constraint

$$\sum_{i=1}^{k} x_{j_i} + \sum_{i=1}^{l}(1 - x_{j_k+i}) \geq 1, \forall j \in \{1, 2, \dots, m\} \tag{2.20}$$

This way we construct a binary IP instance whose constraint j correspond to the requirement that clause c_j must be satisfied, i.e., at least one of the literals in clause c_j must be satisfied as true. Therefore, if this binary IP instance accepts a feasible solution $\hat{x}$, then we set $t(u_i) = T$ whenever $\hat{x}_i = 1$, and $t(u_i) = 0$ if $\hat{x}_i = 0$. By feasibility of $\hat{x}$, this truth assignment t satisfies all the clauses $c_j \in C$ since by construction (from the fact that $\hat{x}$ satisfies (2.20)), at least one of the elements of each c_j is true.

[3] In the definition of reduction given above, this would be ρ.
[4] In the definition of reduction given above, this would be $\phi(\rho)$.

Conversely, if there is a satisfying truth assignment t for C, we let $\hat{x}_i = 1$ if $t(u_i) = T$ and $\hat{x}_i = 0$ if $t(u_i) = F$. Since t is satisfying assignment, for all clauses $c_j \in C$ at least one literal $u \in c_j$ is such that $t(u) = T$, which translates by construction of $\hat{x}$ that $\hat{x}$ is feasible solution for the system given by (2.20). Therefore, we conclude that the arbitrary SAT instance is satisfiable $\iff$ the constructed particular instance of binary IP is feasible.

Lastly, we note that the size of the particular instance of binary IP we constructed has n decision variables (one for each literal) and m constraints (one for each clause). Hence its size its clearly $O(mn)$ (the size of the matrix $[A \;\; b]$), and thus the polynomial reduction of SAT to binary IP is complete. This means that SAT $\preceq$ binary IP, i.e., binary IP is an $\mathcal{NP}$ problem (this was proved at the beginning of this proof) that is at least as difficult as the SAT problem. But SAT is $\mathcal{NPC}$, meaning that SAT is at least as difficult as all the $\mathcal{NP}$ problems. Then using transitivity, binary IP is at least as difficult as any other $\mathcal{NP}$ problem, following that binary IP is $\mathcal{NPC}$. This completes the proof. $\square$

Consider the variant of the SAT problem called 3SAT, whose setting is similar to the one of the SAT problem except for one difference: each clause $c_j \in C$ has *exactly* 3 literals.

2.3.5 3SAT

Theorem 2.4. *The 3-SAT problem is $\mathcal{NP}$-Complete.*

Proof. We need to prove two things:

1. 3-SAT is in $\mathcal{NP}$.

   Proof. Given any truth assignment for all literals, we can then verify all of the clauses in linear time. Therefore the 3-SAT problem is in NP. $\square$

2. SAT $\preccurlyeq$ 3-SAT, i.e. given an instance of SAT (parameterized by (U, C), where U is the set of literals, and C the set of clauses), we can construct an instance of 3-SAT (parameterized by $(U'\ ,C'\ )$) such that:

   (a) (U, C) is satisfiable in SAT $\iff$ (U', C') is satisfiable in 3-SAT.

   (b) The size of the instance of (U', C') is polynomially bounded by the size of the instance (U, C).

We will now present the construction method. Let (U, C) be any instance of SAT, with $|C| = m$. For every clause $C_j \in C$ we will construct a set of clauses $C'^j \subseteq C'$. Each clause in C'^j will have literals from U and additional literals from the set U'_j that will be used only in clauses inside C'^j. So we have

$$U' = U \cup \left(\bigcup_{j=1}^{m} U'_j \right)$$
$$C' = \bigcup_{j=1}^{m} C'^j$$

The construction of U'_j and C'^j will depend on the number of literals k in clause $C_j = \{z_1, \dots, z_k\}$.

- Case 1: $k = 1$, $C_j = \{z_1\}$. Then

$$\begin{aligned} U'_j &:= \{y_1^j, y_2^j\} \\ C'^j &:= \{\{z_1, y_1^j, y_2^j\}, \{z_1, \bar{y}_1^j, y_2^j\}, \{z_1, y_1^j, \bar{y}_2^j\}, \{z_1, \bar{y}_1^j, \bar{y}_2^j\}\} \end{aligned}$$

- Case 2: $k = 2$, $C_j = \{z_1, z_2\}$. Then

$$\begin{aligned} U'_j &:= \{y_1^j\} \\ C'^j &:= \{\{z_1, z_2, y_1^j\}, \{z_1, z_2, \bar{y}_1^j\}\} \end{aligned}$$

- Case 3: $k = 3$, $C_j = \{z_1, z_2, z_3\}$. Then

$$\begin{aligned} U'_j &:= \emptyset \\ C'^j &:= \{\{z_1, z_2, z_3\}\} \end{aligned}$$

- Case 4: $k \geq 4$, $C_j = \{z_1, \dots, z_k\}$. Then

$$\begin{aligned} U'_j :=& \{y_1^j, y_2^j, \dots, y_{k-3}^j\} \\ C'^j :=& \{\{z_1, z_2, y_1^j\}, \{\bar{y}_1^j, z_3, y_2^j\}, \dots, \\ & \{\bar{y}_q^j, z_{q+2}, y_{q+1}^j\}, \dots, \{\bar{y}_{k-4}^j, z_{k-2}, y_{k-3}^j\}, \{\bar{y}_{k-3}^j, z_{k-1}, z_k\}\} \end{aligned}$$

Proof of 2(a). We will first prove that if (U, C) is satisfiable in SAT $\implies$ (U', C') is satisfiable in 3-SAT.
Given a solution that satisfies the (U, C) SAT instance, let $t : U \rightarrow$

$\{T, F\}$ (true and false respectively) be a function that returns the assignment of literals in U of such SAT solution. We will now prove there exists an assignment function $t' : U' \to \{T, F\}$ that satisfies the 3-SAT instance (U', C'). We will do this by proving that for each clause $C_j \in C$ satisfied by t we can also satisfy all clauses $C'^j \subseteq C'$ with an assignment t' defined by:

- Case 1: $k = 1$, $C_j = \{z_1\}$. We set $t'(z_1) = t(z_1) = T$ (we know z_1 must be true trivially). $t'(y_1^j)$ and $t'(y_2^j)$ can be assigned arbitrarily. Trivially all clauses in C'^j will be satisfied by t'.
- Case 2: $k = 2$, $C_j = \{z_1, z_2\}$. We set $t'(z_1) = t(z_1)$ and $t'(z_2) = t(z_2)$. $t'(y_1^j)$ can be assigned arbitrarily. All clauses in C'^j will be satisfied by t' since we know either $t(z_1) = T$ or $t(z_2) = T$.
- Case 3: $k = 3$, $C_j = \{z_1, z_2, z_3\}$. We set $t'(z_1) = t(z_1)$, $t'(z_2) = t(z_2)$, and $t'(z_3) = t(z_3)$ which trivially guarantees the only clause in C'^j will be satisfied by t'.
- Case 4: $k \geq 4$, $C_j = \{z_1, \ldots, z_k\}$. We first set $t'(z_i) = t(z_i), \forall i \in \{1, \ldots, k\}$. We know there exists $l \in \{1, \ldots, k\}$ such that $t(z_l) = T$.
  - If $l \in \{1, 2\}$ then set $t'(y_i^j) = F, \forall i \in \{1, \ldots, k-3\}$. We can see all clauses in C'^j will be satisfied by t' since the first clause is already satisfied by z_l, and all the others have a $\bar{y}_i^j$ literal.
  - If $l \in \{3, \ldots, k-2\}$ then set $t'(y_i^j) = T, \forall i \in \{1, \ldots, l-2\}$ and $t'(y_i^j) = F$, $\forall i \in \{l-1, \ldots, k-3\}$. We can see all clauses in C'^j will be satisfied by t' since clause containing z_l is already satisfied, all the previous clauses have a y_i^j literal, and all the other clauses (after the clause containing z_l) have a $\bar{y}_i^j$ literal.
  - If $l \in \{k-1, k\}$ then set $t'(y_i^j) = T, \forall i \in \{1, \ldots, k-3\}$. We can see all clauses in C'^j will be satisfied by t' since the last clause is already satisfied by z_l, and all the others have a y_i^j literal.

Now we will prove that if (U', C') is satisfiable in 3-SAT $\implies$ (U, C) is satisfiable in SAT.
We have a given assignment t' that satisfies the (U', C') 3-SAT instance. We will build an assignment t that satisfies the (U, C) SAT instance by having the same assignments as t' for all $z_i, i \in \{1, \ldots, k\}$.

We will prove that for each set of clauses $C'^{j} \in C'$ satisfied by t' we also satisfy clause $C_j \subseteq C$ with assignment t:

- Case 1: $k = 1$, $C_j = \{z_1\}$. Is trivial to see that $t'(z_1) = T = t(z_1)$ so C_j is satisfied.
- Case 2: $k = 2$, $C_j = \{z_1, z_2\}$. Either $t'(z_1) = T = t(z_1)$ or $t'(z_2) = T = t(z_2)$ so C_j is satisfied.
- Case 3: $k = 3$, $C_j = \{z_1, z_2, z_3\}$. Trivially C_j is satisfied.
- Case 4: $k \geq 4$, $C_j = \{z_1, \ldots, z_k\}$. We claim that for some $l \in \{1, \ldots, k\}$ we have that $t'(z_l) = T = t(z_l)$, and so C_j is satisfied. Suppose for contradiction that $t'(z_l) = F, \forall l \in \{1, \ldots, k\}$. By the first clause and induction we have $t'(y_i^j) = T, \forall i \in \{1, ..., k-3\}$. However, this implies that the last clause in C'^{j} is not satisfied by t' , which is the required contradiction.

□

Proof of 2(b). Let $|C| = m$, and $|U| = n$. For every $C_j \in C$ the number of new variable is at most $|C_j| + 1 \leq n + 1$, and the total number of clauses in C'^{j} is at most $|C_j| + 3 \leq n + 3$. Therefore for the number of literals we have $|U'| \leq n + m(n+1)$, and for the number of clauses $|C'| \leq m(n+3)$ respectively. Thus, the new 3-SAT instance is bounded polynomially in the size of the original SAT instance. □

Thus we finished proving SAT $\preccurlyeq$ 3-SAT, therefore 3-SAT is $\mathcal{NP}$-Complete.

□

2.3.6 Vertex-Cover Problem

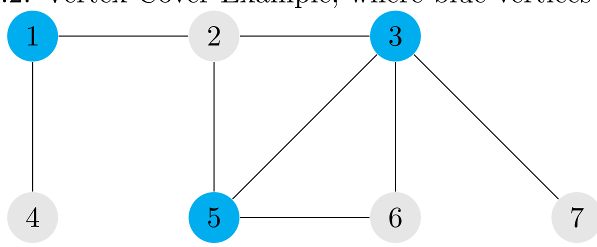
Definition 2.13 (Vertex-Cover)**.** *A vertex-cover of an undirected graph $G = (V, E)$ is a subset V' of V such that if edge (u, v) is an edge of G then either $u \in V'$ or $v \in V'$ (or both).*

Definition 2.14 (Vertex-Cover Problem)**.** *Given $G = (V, E)$ and a positive integer k, the goal is to determine a vertex-cover with size of at most k.*

Theorem 2.5. *The Vertex-Cover Problem is in $\mathcal{NP}$-Complete.*

Proof. 1. We first verify that the Vertex-Cover Problem is in $\mathcal{NP}$.

Figure 2.2: Vertex-Cover Example, where blue vertices are in V'



Proof. For each vertex in V, remove all incident edges and check if all edges were removed from E. This can be checked in at most $|E|+|V|$ time. Thus, the Vertex-Cover Problem is in $\mathcal{NP}$. □

2. We then prove that 3-SAT $\preccurlyeq$ Vertex-Cover Problem, i.e. given an instance of 3-SAT (parameterized by (U,C)), we can construct an instance of the Vertex-Cover Problem (parameterized by $G=(V,E)$ and k) such that:

   (a) (U,C) is satisfiable in 3-SAT $\iff$ $\{G=(V,E),k\}$ is satisfiable in the Vertex-Cover Problem.

   (b) The size of the instance $\{G=(V,E),k\}$ is polynomially bounded by the size of the instance (U,C).

   To construct $\{G=(V,E),k\}$ we start by including vertices into V:

   - For every literal $U_i \in U$ we construct two vertices V_i and $\bar{V}_i$.
   - For every clause $C_j \in C$ we construct three vertices C_j^1, C_j^2, and C_j^3.

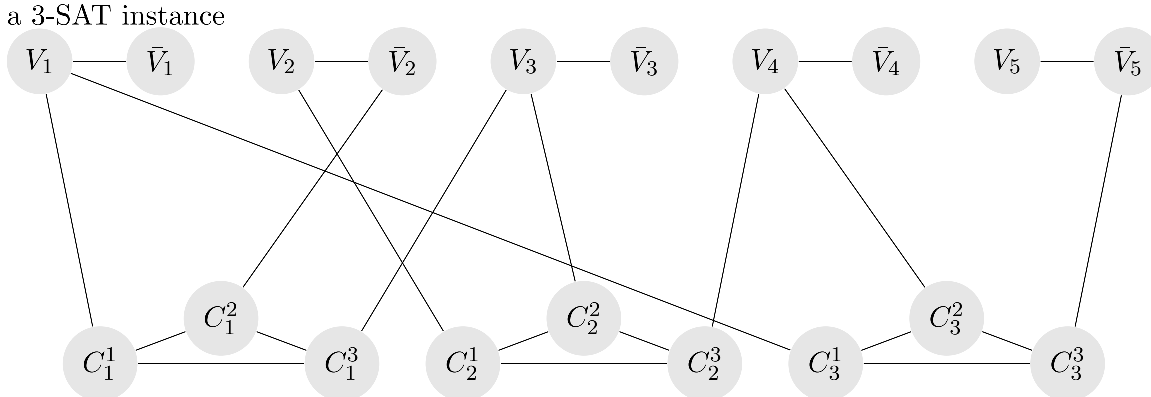
   Then we include the following edges into E:

   - $(V_i, \bar{V}_i), \forall i \in \{1,\ldots,|U|\}$.
   - $(C_j^1, C_j^2), (C_j^2, C_j^3), (C_j^3, C_j^1), \forall j \in \{1,\ldots,|C|\}$.
   - For every clause $C_j = \{z_1, z_2, z_3\} \in C$, and for every $t \in \{1,2,3\}$: include edge (C_j^t, V_i) if $z_t = V_i$ or include edge $(C_j^t, \bar{V}_i)$ if $z_t = \bar{V}_i$.

   We then set $k = n + 2m$.

   On Figure 2 we can see a construction example for $U = \{U_1, U_2, U_3, U_4, U_5\}$ and $C = \{\{U_1, \bar{U}_2, U_3\}, \{U_2, U_3, U_4\}, \{U_1, U_4, \bar{U}_5\}\}$.

Figure 2.3: Construction example of a Vertex-Cover Problem instance from a 3-SAT instance



Proof of 2(a). Let $|U| = n$ and $|C| = m$. We will first prove that (U, C) is satisfiable in 3-SAT $\implies$ $\{G = (V, E), k\}$ is satisfiable in the Vertex-Cover Problem.
Let $t : U \to \{T, F\}$ be a function that returns the assignment of literals in U that satisfy the 3-SAT instance. We will now prove there exists a vertices set V' that satisfies the instance $\{G = (V, E), k\}$ of the Vertex-Cover Problem.

Include in V' all vertices that correspond to true literals, i.e., if $t(U_i) = T$ then include V_i in V' and if $t(U_i) = F$ then include $\bar{V}_i$. Since the 3-SAT instance is satisfiable for clause $C_j = \{z_1, z_2, z_3\}$, then there exists $l \in \{1, 2, 3\}$ such that $t(z_l) = T$; then include vertices C_j^t in V' for $t \in \{1, 2, 3\}\backslash\{l\}$.

Note that the constructed V' is of size k. Moreover, it is easily verified that V' is a vertex-cover.

Now we will prove that $\{G = (V, E), k\}$ is satisfiable in the Vertex-Cover Problem $\implies$ (U, C) is satisfiable in 3-SAT.
We have a given vertex-cover V' that satisfies the Vertex-Cover Problem instance with $|V'| \leq k$. We will build an assignment t that satisfies the (U, C) 3-SAT instance.

For every edge $(V_i, \bar{V}_i), \forall i \in \{1, \ldots, |U|\}$ we know at least one if its vertices is included in V'. For every edges of the form

$$(C_j^1, C_j^2), (C_j^2, C_j^3), (C_j^3, C_j^1), \forall j \in \{1, \ldots, |C|\}$$

we know at least two of the vertices C_j^1, C_j^2, C_j^3 is in V'. Thus $|V'| \geq k = n + 2m$, and since we know $|V'| \leq k$, then $|V'| = k$. This means that exactly one of the V_i or $\bar{V}_i$ is in the vertex-cover $V', \forall i \in \{1, \ldots, |U|\}$; and also exactly two of C_j^1, C_j^2, C_j^3 are in the vertex-cover $V', \forall j \in \{1, \ldots, |C|\}$. Then we set the assignment as:

$$t(U_i) = \begin{cases} T & \text{if } V_i \in V' \\ F & \text{if } \bar{V}_i \in V' \end{cases}$$

It is now straight forward to verify that the 3-SAT instance is satisfiable. □

Proof of 2(b). The number of vertices of the instance $\{G = (V, E), k\}$ of the Vertex-Cover Problem is given by $|V| = 2n + 3m$, the number of edges is $|E| = n + 3m + 3m$, and $k = n + 2m$. Thus, the instance of the Vertex-Cover Problem is bounded polynomially in the size of the original 3-SAT instance. □

Thus we finished proving 3-SAT $\preccurlyeq$ Vertex-Cover Problem, therefore the Vertex-Cover Problem is $\mathcal{NP}$-Complete.

□

2.4 Suggested exercises

I reccomend that you read all the worked-out examples and excercises in chapter I.5 of textbook.

Chapter 3

Review of Polyhedral Theory

3.1 Basic Definitions and Results

Definition 3.1 (Hyperplane)**.** *A hyperplane is a set defined by a single equality. Precisely, a hyperplane in* $\mathbb{R}^n$ *is a set of the form*

$$H = \{x \in \mathbb{R}^n \mid a^\top x = b\},$$

where $a \in \mathbb{R}^n, b \in \mathbb{R}$*. It separates the space into two half-spaces.*

Definition 3.2 (Half-space)**.** *A half-space is a set defined by a single affine inequality. Precisely, a half-space in* $\mathbb{R}^n$ *is a set of the form*

$$H = \{x \in \mathbb{R}^n \mid a^\top x \leq b\},$$

where $a \in \mathbb{R}^n, b \in \mathbb{R}$*. The boundary of a half-space is a hyperplane.*

Definition 3.3 (Polyhedron)**.** *The intersection of a finite number of half-spaces is a polyhedron.*

Since half-spaces are convex sets, therefore polyhedron is also a convex set.

Definition 3.4 (Polytope)**.** *A bounded polyhedron is a polytope.*

3.1.1 Fourier-Motzkin Projection

Definition 3.5. *Let* $X \subseteq \mathbb{R}^n$*. Then the* projection *of* X *onto the first* $n-1$ *components is*

$$\mathrm{Proj}_{x_1,\ldots,x_{n-1}}(X) = \{(x_1,\ldots,x_{n-1}) \in \mathbb{R}^{n-1} | \exists x_n \in \mathbb{R} (x_1,\ldots,x_{n-1},x_n) \in X\}.$$

Theorem 3.1 (Fourier-Motzkin)**.** *Let $P \subseteq \mathbb{R}^n$ be a polyhedron. Then $\mathrm{Proj}_{x_1,\ldots,x_{n-1}}(P)$ is a polyhedron.*

Proof. Suppose $P = \{x | a_i^\top x \leq b_i, i \in \{1, \ldots, m\}\}$. A vector x is contained in P if and only if

$$a_{in} x_n \leq b_i - \bar{a}_i{}^\top \bar{x},$$

where $\bar{v}$ is the projection of v onto the first $n-1$ coordinates. Let $I_1 = \{i | a_{in} > 0\}$, $I_2 = \{i | a_{in} < 0\}$, and $I_3 = \{i | a_{in} = 0\}$. In these three cases we can simplify the inequalities by dividing through by a_{in} to get

$$x_n \leq \hat{b}_i - \hat{a}_i{}^\top \bar{x}, \quad x_n \geq \hat{b}_i - \hat{a}_i{}^\top \bar{x}, \text{ and } 0 \leq \hat{b}_i - \hat{a}_i{}^\top \bar{x}, \tag{3.1}$$

respectively. Let $Q \subseteq \mathbb{R}^{n-1}$ be the set of points satisfying

$$\hat{b}_j - \hat{a}_j \bar{x} \geq \hat{b}_k - \hat{a}_k \bar{x} \quad \forall j \in I_1, k \in I_2 \tag{3.2}$$

$$0 \leq \hat{b}_i - \hat{a}_i \bar{x} \qquad \forall i \in I_3 \tag{3.3}$$

We will show that $Q = \mathrm{Proj}_{x_1,\ldots,x_{n-1}}(P)$. Indeed, for any $\bar{x} \in \mathrm{Proj}(P)$ there exists x_n such that $(\bar{x}, x_n) \in P$. Then $\bar{x}$ satisfies the system of inequalities for Q, and thus $\bar{x} \in Q$. Conversely, for any $\bar{x} \in Q$,

$$\min_{j \in I}\{\hat{b}_j - \hat{a}_j \bar{x}\} \geq \max_{k \in I_2}\{\hat{b}_k - \hat{a}_k \bar{x}\},$$

and so we can pick x_n between them. Then $(\bar{x}, x_n)$ satisfies the conditions of all types for P, and is thus $\bar{x} \in \mathrm{Proj}(P)$. □

Proposition 3.1. *The use of the term "*max*" is justified for the LP problem*

$$z^* = \sup\{c^\top x \mid Ax \leq b\}$$

(assuming the supremum exists.)

Proof. Let $Q = \{(x, z) \in \mathbb{R}^{n+1} | Ax \leq b,\ z - c^\top x \leq 0\}$. Then z^* is the largest value of z for which there is an x such that $(x, z) \in Q$. Project out all of the x variables to get Q'. Then $z^* = \sup_{z \in Q'} z$ where Q' is a 1-dimensional polyhedron, i.e. a closed interval. Thus if the supremum exists it is actually obtained by some $z \in Q'$. □

The Fourier-Motzkin procedure can be use to prove the following fundamental results:

Theorem 3.2 (Farkas' Lemma). *Exactly one of the following systems has a solution:*

$$\begin{aligned} &Ax = b, x \geq 0, \\ &y^\top A \leq 0, y^\top b > 0. \end{aligned}$$

Theorem 3.3 (Farkas-Minkowski-Weyl Theorem). *A convex cone is a polyhedron if and only if it is finitely generated. Therefore*

$$\{x \in \mathbb{R}^n \mid Ax \leq 0\} = \left\{ x = \sum_{i=1}^{k} \lambda^i x^i : \lambda^i \geq 0,\ \forall i \in \{1, 2, , \ldots, k\} \right\},$$

where $|k| < \infty$.

Theorem 3.4 (Decomposition Theorem). *A set P is a polyhedron if and only if $P = Q + C$, where Q is a polytope and C is a polyhedral cone.*

3.1.2 Duality Theory for Linear Programming

Given a maximization (resp. minimization) linear program, a feasible solution gives a lower bound (resp. upper bound) to the optimal objective function value. For every linear program, there is an associated linear program which can be used to obtain bounds in other direction, the so -called dual bound. This associated linear program is called a dual program. For a minimization (resp. maximization) program the dual gives a lower (resp. upper) bound on the optimal objective function. Next the general construction of a dual is presented.

Definition 3.6.

1. *Given a linear program with m constraints and n variables, the dual has one variable for each constraint of the linear program (we will often call the starting linear program as the <u>primal linear program</u>) and one constraint for each variable of the primal, i.e. the dual has n constraints and m variables.*

2. *If the primal is a minimization problem, then the dual is a maximization problem. Similarly if the primal is a maximization problem, then the dual is a minimization problem.*

3. *The objective coefficients of the primal become the right-hand-sides of the dual and the right-hand-sides of the primal become the objective coefficients of the dual.*

4. *Each column in the left-hand-side of the primal constraints matrix (i.e. coefficients corresponding to a particular variable) becomes a row (i.e. coefficients of a single constraint) in the left-hand-side of the dual constraints matrix and vice-verse.*

5. *Based on whether the primal is min or a max problem, there is an exact relationship between*
   (A) sign of constraint in primal$\leftrightsquigarrow$ type of corresponding variable in dual
   and
   (B) type of variable in primal$\leftrightsquigarrow$sign of corresponding constraint in dual.

For a minimization problem, the dual is found as follows. Consider first the primal LP:

$$\min \quad \sum_{j=1}^{n} c_j x_j \tag{3.4}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad \text{for all } i = 1, ..., m_1 \tag{3.5}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad \text{for all } i = m_1 + 1, ..., m_2 \tag{3.6}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for all } i = m_2 + 1, ..., m \tag{3.7}$$

$$x_j \geq 0 \quad \text{for all } j = 1, ..., n_1 \tag{3.8}$$

$$x_j \text{ is free} \quad \text{for all } j = n_1 + 1, ..., n_2 \tag{3.9}$$

$$x_j \leq 0 \quad \text{for all } j = n_2 + 1, ..., n. \tag{3.10}$$

Then the dual is:

$$\max \quad \sum_{i=1}^{m} b_i y_i \tag{3.11}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} a_{ij} y_i \le c_j \quad \text{for all } j = 1, ..., n_1 \tag{3.12}$$

$$\sum_{i=1}^{m} a_{ij} y_i = c_j \quad \text{for all } j = n_1 + 1, ..., n_2 \tag{3.13}$$

$$\sum_{i=1}^{m} a_{ij} y_i \ge c_j \quad \text{for all } i = n_2 + 1, ..., n \tag{3.14}$$

$$y_i \ge 0 \quad \text{for all } i = 1, ..., m_1 \tag{3.15}$$

$$y_i \text{ is free} \quad \text{for all } i = m_1 + 1, ..., m_2 \tag{3.16}$$

$$y_i \le 0 \quad \text{for all } i = m_2 + 1, ..., m. \tag{3.17}$$

The dual of a dual is the primal. So if the dual above (i.e. (3.11) - (3.17)) was the original LP, then (3.4) - (3.10) would be its dual. □

Example 3.1. *Primal:*

$$\max \quad x_1 + 2x_2 + 3x_3 \tag{3.18}$$

$$\text{s.t.} \quad 8x_1 + 9x_2 + 10x_3 \ge 4 \tag{3.19}$$

$$11x_1 + 12x_2 + 13x_3 = 5 \tag{3.20}$$

$$14x_1 + 15x_2 + 16x_3 \le 6 \tag{3.21}$$

$$17x_1 + 18x_2 + 19x_3 \le 7 \tag{3.22}$$

$$x_1 \ge 0 \tag{3.23}$$

$$x_2 \text{ is free} \tag{3.24}$$

$$x_3 \le 0. \tag{3.25}$$

Then the dual is:

$$\min \quad 4y_1 + 5y_1 + 6y_3 + 7y_4 \tag{3.26}$$

$$\text{s.t.} \quad 8y_1 + 11y_2 + 14y_3 + 17y_4 \ge 1 \tag{3.27}$$

$$9y_1 + 12y_2 + 15y_3 + 18y_4 = 2 \tag{3.28}$$

$$10y_1 + 13y_2 + 16y_3 + 19y_4 \le 3 \tag{3.29}$$

$$y_1 \le 0 \tag{3.30}$$

$$y_2 \text{ is free} \tag{3.31}$$

$$y_3, y_4 \ge 0. \tag{3.32}$$

To clarify again, the rules for the the relationship between the sign of the constraints and that of the corresponding variable in the dual are the following:

max	min
Constraint type	Variable type
"$\leq$"	Variable is non-negative "≥ 0"
"$\geq$"	Variable is non-positive "≤ 0"
"$=$"	Variable is free
Variable type	Constraint type
Variable is non-negative "≥ 0"	"$\geq$ "
Variable is non-positive "≤ 0 "	"$\leq$"
Variable is free	"$=$"

The above table works both ways. That is if we have a max primal, then to construct the dual go from left to right. If we have a min primal, then to construct the dual go from right to left.

Proposition 3.2 ('Weak Duality' aka the bounding result)**.** *If $\hat{x}$ is any feasible solution of minimization (resp. maximization) primal linear problem whose objective function is $c^T x$ and $\hat{y}$ is any feasible solution of dual maximization (resp. minimization) linear problem whose objective function is $b^T y$, then $c^T \hat{x} \geq b^T \hat{y}$ (resp. $c^T \hat{x} \leq b^T \hat{y}$).*

There is an important consequence of Proposition 3.2, that we present next.

Proposition 3.3. *If the primal is unbounded, then the dual is infeasible.*

The weak dual result says that if we write the dual of a maximization problem (resp minimization problem), then any feasible solution of the dual gives an upper bound (resp. lower bound) to the optimal objective function value of the primal. We would like to next know how good is this bound? It turns out the bound is tight in the sense that optimal objective function value of the primal and dual have exactly the same value. Formally the following statement can be proven.

Proposition 3.4 ('Strong Duality Result' aka best dual bound is tight)**.** *Suppose that the primal LP (whose objective function is $c^T x$) has an optimal solution x^*, then the dual LP (whose objective function is $b^T y$) has an optimal solution y^* such that $c^T x^* = b^T y^*$. Moreover, the converse is also true. Therefore,*

1. *The primal LP has an optimal solution if and only if the dual has an optimal solution.*

2. *In the case of existence of optimal solutions, the optimal objective function values of the primal and dual are equal.*

If the primal is infeasible, then by the fact that the dual of a dual is the primal and Proposition 3.3 intuitively it appears the dual may be unbounded. It turns out that the dual may also be infeasible. Here is a classical example:

Example 3.2. *Consider the LP:*

$$\begin{aligned} \mathit{max} \quad & 2x_1 - x_2 && (3.33)\\ \mathit{s.t.} \quad & x_1 - x_2 \leq 1 && (3.34)\\ & -x_1 + x_2 \leq -2 && (3.35)\\ & x_1, x_2 \geq 0 && (3.36) \end{aligned}$$

and its dual:

$$\begin{aligned} \mathit{min} \quad & y_1 - 2y_2 && (3.37)\\ \mathit{s.t.} \quad & y_1 - y_2 \geq 2 && (3.38)\\ & -y_1 + y_2 \geq -1 && (3.39)\\ & y_1, y_2 \geq 0. && (3.40) \end{aligned}$$

Both the primal and dual are infeasible. □

Now we are ready to give the complete relationship between a primal and its dual. This is illustrated next. A ✓ implies the combination is possible, while × implies that the combination is not possible.

	Dual Optimal	Dual Infeasible	Dual Unbounded
Primal Optimal	✓	×	×
Primal Infeasible	×	✓	✓
Primal Unbounded	×	✓	×

The first row and column in table above are consequence of Proposition 3.4. The entries $(2, 3)$, $(3, 2)$, $(3, 3)$ in the table above are illustrated/consequence of Proposition 3.3. Example 3.2 is an illustration of entry $(2, 2)$.

A very important consequence of the duality theorem is the so called Complementary Slackness result. We next present this result.

Proposition 3.5 (Complementary Slackness). *Let $x^* \in \mathbb{R}^n$ and $y^* \in \mathbb{R}^m$ be primal and dual feasible respectively. Then x^* and y^* are optimal solutions for the primal and dual problems respectively if and only if they satisfy the following conditions:*

1. *Primal complementary conditions: For each constraints of the primal - either the constraint is tight, i.e. $\sum_{j=1}^{n} a_{ij}x_j^* = b_i$ or the corresponding dual variable $y_i^* = 0$.*

2. *Dual complementary conditions: For each constraints of the dual - either the constraint is tight, i.e. $\sum_{i=1}^{m} a_{ij}y_i^* = c_j$ or the corresponding primal variable $x_j^* = 0$.* □

3.2 Recession Cone, Linearity Space of a Polyhedron

Definition 3.7 (Direction of Recession). *Given a nonempty convex set C, a vector d is a direction of recession at a point $x^0 \in C$ if $x^0 + \lambda d \in C$ for all $\lambda \geq 0$.*

It is straightforward to see that the set of recession directions at a point x^0 for a set C forms a cone[1]

Proposition 3.6. *If C is a closed convex set, the set of recession directions is identical at each point in C.*

Proof. It is sufficient to show that d is recession direction at $x \in C$, then d is recession direction at $y \in C$. In particular set $\lambda_0 \geq 0$. We will show that $\lambda_0 d + y \in C$. Observe that, for any $1 > \epsilon > 0$

$$C \ni \epsilon\left(x + \frac{\lambda_0}{\epsilon}d\right) + (1-\epsilon)y = (\epsilon x + (1-\epsilon)y) + \lambda_0 d,$$

where the first containment is due to convexity of C. As $\epsilon \to 0$, we have that $(\epsilon x + (1-\epsilon)y) + \lambda_0 d$ approaches $y + \lambda_0 d$. By closedness of C, $y + \lambda_0 d \in C$. □

Since a polyhedron is a closed convex set, it makes sense to talk about the recession cone of the polyhedron (since the recession cone for any point is identical). Therefore, we obtain the following definition.

[1]A cone is set that is closed under multiplication with non-negative scalars, i.e C is cone iff $x \in C$ implies $\lambda x \in C$ for all $\lambda \geq 0$.

Definition 3.8 (Recession Cone)**.** *For convex set $S \subseteq \mathbb{R}^n$, the recession cone (denoted as* rec.cone(S)*) is given by*

$$\text{rec.cone}(S) = \{d \in \mathbb{R}^n \mid x + \lambda d \in S,\ \forall \lambda \geq 0, \forall x \in S\}.$$

Proposition 3.7. *If $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, then* rec.cone$(P) = \{x \in \mathbb{R}^n \mid Ax \leq 0\}$.

Proof. We first prove rec.cone$(P) \supseteq \{x \in \mathbb{R}^n \mid Ax \leq 0\}$. Let $x \in P$ and $d \in \{u \in \mathbb{R}^n \mid Au \leq 0\}$. Then for all $\lambda \geq 0$, $A(x+\lambda d) = Ax + \lambda Ad \leq Ax \leq b$, where the first inequality uses the fact that $\lambda \geq 0$ and $Ad \leq 0$ and the last inequality follows from the definition of P.

Now we show rec.cone$(P) \subseteq \{x \in \mathbb{R}^n \mid Ax \leq 0\}$. Let $x \in P$ and $d \in$ rec.cone(P). Assume by contradiction, $Ad \not\leq 0$, i.e. the ith component of the vector $(Ad)_i > 0$. Then $(A(x + \lambda d))_i = (Ax)_i + \lambda(Ad)_i > b_i$ for sufficiently large λ, which is a contradiction since $(A(x + \lambda d))_i \leq b$. □

Definition 3.9 (Linearity Space)**.** *For a given convex set $S \subseteq \mathbb{R}^n$, the linearity space is the set* lin.space$(S) = \{d \in \mathbb{R}^n \mid x + \lambda d \in S,\ \forall \lambda \in \mathbb{R}, \forall x \in S\}$.

The linearity space of a convex set is the set of lines in the set.

Definition 3.10 (Pointed Polyhedron)**.** *A polyhedron is pointed if* lin.space$(P) = \{0\}$.

A nonempty polyhedron is pointed when it does not contain any line.

Proposition 3.8. *For $P = \{x \in \mathbb{R}^n \mid Ax \leq b\} \subseteq \mathbb{R}^n$,* lin.space$(P) = \{x \in \mathbb{R}^n \mid Ax = 0\}$.

Proof. By definition, lin.space(P) = rec.cone$(P) \cap$ rec.cone$(-P)$. Since rec.cone$(P) = \{x \in \mathbb{R}^n \mid Ax \leq 0\}$, we have that lin.space$(P) = \{x \in \mathbb{R}^n \mid Ax = 0\}$. □

3.3 Dimension of a Polyhedron

Definition 3.11 (Affine Combination)**.** *A point $x \in \mathbb{R}^n$ is an affine combination of $x^1, \ldots, x^t \subseteq \mathbb{R}^n$ if there exists scalars $\lambda_1, \ldots, \lambda_t$ such that $\sum_{i=1}^t \lambda_i x^i$ and $\sum_{i=1}^t \lambda_i = 1$.*

What does it mean for ten points to be affinely independent? No point of these ten can be expressed as an affine combination of the other nine points.

Definition 3.12 (Affine Independent). *A set of points $x^1, \ldots, x^k \in \mathbb{R}^n$ are said to be affinely independent if none of the points can be written as an affine combination of the other vectors.*

Proposition 3.9. *A set of points $x^1, \ldots, x^k \in \mathbb{R}^n$ are affinely independent if and only if the unique solution to $\sum_{i=1}^k \alpha_i x^i = 0$ and $\sum_{i=1}^k \alpha_i = 0$ is $a_i = 0, \forall i = 1, \ldots, k$.*

Proof. We first prove the 'if' part. Assume without loss of generality, $x^1 = \sum_{i=2}^k \lambda_i x^i$ where $\sum_{i=2}^k \lambda_i = 1$. Then $\alpha_1 = -1$ and $\alpha_i = \lambda_i$ for $i \in \{2, \ldots, k\}$ is a nonzero solution to $\sum_{i=1}^k \alpha_i x^i = 0$ and $\sum_{i=1}^k \alpha_i = 0$.

To show the converse, assume that $\sum_{i=1}^k \alpha_i x^i = 0$ and $\sum_{i=1}^k \alpha_i = 0$ has a nonzero solution. Without loss of generality assume that $\alpha_1 \neq 0$. Then we have that $x^1 = \sum_{i=2}^k -\alpha_i/\alpha_1 x^i$ where $\sum_{i=2}^k -\alpha_i/\alpha_1 = 1$. □

Linear independence implies affine independence but the converse is not true.

Observation 3.1. *The maximum number of affinely independent points in $\mathbb{R}^n$ is $n + 1$.*

Proposition 3.10. *Given $x^1, \ldots, x^m \in \mathbb{R}^n$, the following are equivalent:*

(i) $x^1, \ldots, x^m$ *are affinely independent;*

(ii) $x^2 - x^1, x^3 - x^1, \ldots, x^m - x^1$ *are linearly independent;*

(iii) $\begin{bmatrix} x^1 \\ 1 \end{bmatrix}, \begin{bmatrix} x^2 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} x^m \\ 1 \end{bmatrix}$ *are linearly independent.*

Definition 3.13 (Affine Subpace/Affine Set). *A subset $\mathcal{A}$ of $\mathbb{R}^n$ is an affine space if $\mathcal{A}$ is closed under taking affine combinations.*

An affine subspace of $\mathbb{R}^n$ is a translated linear subspace. The linear subspaces are precisely the affine subspaces containing the origin. An affine set is represented by $Ax = b$ while a linear subspace may be represented as $Ax = 0$.

Definition 3.14 (Affine Hull). *The inclusionwise minimal affine space containing a set $S \in \mathbb{R}^n$ is called the affine hull of S and is denoted as* $\text{aff.hull}(S)$.

Definition 3.15 (Dimension). *The dimension of a polyhedron P is the dimension of its affine hull and is denoted as* $\dim(P)$.

We say a polyhedron $P := \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}, A^{\leq}x \leq b^{\leq}\}$ if all the feasible points belonging to the polyhedron satisfy the inequality $A^{=}x \leq b^{=}$ at equality. For every inequality in the set $A^{\leq}x \leq b^{\leq}$, there exists a point in P which does not satisfy this inequality as equality.

Proposition 3.11. *Let $P = \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}, A^{\leq}x \leq b^{\leq}\}$ be a nonempty polyhedron, then there exists $x^0 \in P$ such that $A^{\leq}x^0 < b^{\leq}, A^{=}x^0 = b^{=}$.*

Proof. Let the set of constraints $A^{\leq}x \leq b^{\leq}$ be indexed by $I^{\leq}$. By definition, for each $i \in I^{\leq}$, there is a point $x^i \in P$ such that $a^i x^i < b_i$. It is easy to verify the point $x^0 = 1/|I^{\leq}| \sum_{i \in I^{\leq}} x^i$ satisfies $A^{\leq}x^0 < b^{\leq}$ and $A^{=}x^0 = b^{=}$. □

Proposition 3.12. *Let $P = \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}, A^{\leq}x \leq b^{\leq}\}$ be a nonempty polyhedron, then* $\text{aff.hull}(P) = \{x \in \mathbb{R}^n \mid A^{=}x = b^{=}\} = \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}\}$.

Proof. We first show $\text{aff.hull}(P) \subseteq \{x \in \mathbb{R}^n \mid A^{=}x = b^{=}\}$. Since $P \subseteq \{x \in \mathbb{R}^n \mid A^{=}x = b^{=}\}$, $\text{aff.hull}(P)$ is the smallest affine subspace containing P, and $\{x \in \mathbb{R}^n \mid A^{=}x = b^{=}\}$ is an affine subspace, we obtain this containment.

The containment $\{x \in \mathbb{R}^n \mid A^{=}x = b^{=}\} \subseteq \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}\}$ is trivial.

Finally we show $\{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}\} \subseteq \text{aff.hull}(P)$. Let $x^0 \in \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}\}$. Pick $x' \in P$ such that $A^{=}x' \leq b^{=}, A^{\leq}x' < b^{\leq}$ (we are using the fact that P is nonempty and Proposition 3.11). There are two cases to consider. When $x^0 = x'$, we have $x^0 \in P$, which implies $x^0 \in \text{aff.hull}(P)$. When $x^0 \neq x'$, since $A^{=}x' \leq b^{=}$ and $A^{\leq}x' < b^{\leq}$, there exists a strict convex combination of x^0 and x' denoted by x^2 that satisfies $A^{=}x^2 \leq b^{=}$, $A^{\leq}x^2 \leq b^{\leq}$, $x^2 \neq x'$. In particular, $x^2 \in P$. Therefore $x^0 \in \text{aff.hull}(P)$ since x^0 can be expressed as affine combination of x' and x^2. □

Corollary 3.1. *Let P be a polyhedron such that $P \subseteq \mathbb{R}^n$, $P \neq \emptyset$, $P = \{x \in \mathbb{R}^n \mid A^{=}x \leq b^{=}, A^{\leq}x \leq b^{\leq}\}$. Then $\dim(P) = n - \text{rank}(A^{=})$.*

3.4 Faces of a Polyhedron

Definition 3.16 (Valid Inequality)**.** *An inequality $\pi^\top x \leq \pi^0$ is a valid inequality for the set $S \in \mathbb{R}^n$ if $S \subseteq \{x \in \mathbb{R}^n \mid \pi^\top x \leq \pi^0\}$.*

Definition 3.17 (Face of a Polyhedron)**.** *Let $P \in \mathbb{R}^n$ be a polyhedron. Given a valid inequality $\pi^\top x \leq \pi^0$ for P, we define a face of P as the set $P \cap \{x \in \mathbb{R}^n \mid \pi^\top x = \pi^0\}$.*

Note that a more general definition of a face of a convex set C is the following. A subset $F \subseteq C$ is a face if for all $x \in F$ the following property holds: If $x = \sum_{i=1}^{t} \lambda_i x^i$ where $x^i \in C$ for all $i = 1, \ldots, t$, $\sum_{i=1}^{t} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i = 1, \ldots, t$, then $x^i \in F$ for all $i = 1, \ldots, t$. For polyhedron, these two definitions are identical, although this is not true for general convex sets. We will not work with this definition in this class.

Observation 3.2. *Let $P \subseteq \mathbb{R}^n$ be a polyhedron. Then P is a face of itself.*

Definition 3.18 (Proper Face)**.** *A face of P is called a proper face of P if it is nonempty and is strictly contained in P.*

Proposition 3.13. *Let $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$. Let F be a face of P. Then $F = \{x \in P \mid A'x = b'\}$, where $A'x \leq b'$ is a subsystem of $Ax \leq b$.*

Proof. Let $F = \{x \in P \mid \pi^\top x = \pi^0\}$ where $\pi^\top x \leq \pi^0$ is a valid inequality for P. Equivalently, we have $F = \text{argmax}\{\pi^\top x \mid x \in P\}$. Consider the following linear program and its dual:

$$\begin{array}{lrl} \text{(Primal)} & \pi^0 = \max & \pi^\top x \\ & \text{s.t.} & Ax \leq b \\ \text{(Dual)} & \pi^0 = \min & y^\top b \\ & \text{s.t.} & y^\top A = \pi^\top \\ & & y \geq 0 \end{array}$$

Let y^* be a dual optimal solution. Let y' be the sub-vector of y^* with positive components. Let $A'x \leq b'$ be the subsystem of $Ax \leq b$ corresponding to y'. Now the result is proven by establishing the following claim: $F = \{x \in P \mid A'x = b'\}$. To see why the claim holds, notice

$$\hat{x} \in F \Leftrightarrow \hat{x} \in P \text{ and } \pi^\top \hat{x} = \pi^0 \Leftrightarrow \hat{x} \in P \text{ and } y'^\top A'\hat{x} = y'^\top b' \Leftrightarrow \hat{x} \in P \text{ and } A'\hat{x} = b',$$

where the last equality is obtained using the fact that $y' > 0$ and $A'\hat{x} \leq b'$. $\square$

Corollary 3.2. *For a nonempty polyhedron, the following hold:*

(i) The number of faces of the polyhedron is finite (since number of combinations of a finite number of constraints is finite);

(ii) A face is a polyhedron;

(iii) A face of a face is a face of the original polyhedron.

Definition 3.19 (Facet)**.** *Inclusionwise maximal proper faces of a polyhedron are called facets.*

Maximal face cannot be contained in a bigger face of the polyhedron other than itself. Since it is a proper face, it should be a proper subset of P.

Proposition 3.14. *Let* $P := \{x \in \mathbb{R}^n | A^{\leq} x \leq b^{\leq}, A^{=} x = b^{=}\} \neq \emptyset$ *and* $\exists$ *at least one non-redundant inequality of* $A^{\leq} x \leq b^{\leq}$*. Assume all inequalities in* $A^{\leq} x \leq b^{\leq}$ *are non-redundant. Then the facets of* P *are in bijection with the inequalities in* $A^{\leq} x \leq b^{\leq}$*. In particular,* F *is a facet* $\iff$ $F = \{x \in P | \alpha^T x = \beta\}$*, where* $\alpha^T x \leq \beta$ *is an inequality from* $A^{\leq} x \leq b^{\leq}$*.*

Proof. $\Rightarrow$: Since F is a facet, $F = \{x \in P | A' x = b'\}$ where $A' x \leq b'$ is a subsystem of $A^{\leq} x \leq b^{\leq}$. Let $\alpha^T x \leq \beta$ be one inequality in $A' x \leq b'$. Let $\hat{F} = \{x \in P | \alpha^T x = \beta\}$. Observe that $F \subseteq \hat{F} \subseteq P$.
Since no inequality in $A^{\leq} x \leq b^{\leq}$ is redundant and $\alpha^T x \leq \beta$ is one of them, $\exists x^0 \in P$, such that $\alpha^T x^0 < \beta$. Thus $\hat{F}$ is a proper face of P. As we assume F is a maximal proper face of P, we obtain $F = \hat{F}$.
$\Leftarrow$: Let $\alpha^T \leq \beta$ is an inequality from $A^{\leq} x \leq b^{\leq}$, and let $\hat{A} x \leq \hat{b}$ be the remaining inequalities in $A^{\leq} x \leq b^{\leq}$. We will show that $F = \{x \in P | \alpha^T x = \beta\}$ is non-empty, so that it is indeed a face. By previous lemma, there exits an $x_1 \in P$ such that $A^{=} x_1 = b^{=}, \alpha^T x_1 < \beta, \hat{A} x_1 < \hat{b}$. Since $\alpha^T x \leq \beta$ is not redundant inequality of P, there exists an x_2 such that $A^{=} x_2 = b^{=}, \alpha^T x_2 > \beta, \hat{A} x_2 \leq \hat{b}$. Then there exists $\hat{x}$, that is a convex combination of x_1 and x_2, satisfies $A^{=} x_0 = b^{=}, \alpha^T x_0 = \beta, \hat{A} x_0 < \hat{b}$. This implies F is a nonempty face. □

Corollary 3.3. *Let* $P := \{x \in \mathbb{R}^n | A^{\leq} x \leq b^{\leq}, A^{=} x = b^{=}\} \neq \emptyset$ *and* $\exists$ *at least one non-redundant inequality of* $A^{\leq} x \leq b^{\leq}$*. Assume all inequalities in* $A^{\leq} x \leq b^{\leq}$ *are non-redundant. Let* F *be a facet of* P*. Then* $dim(F) = dim(P) - 1$*.*

Proof. According to Proposition 1, $F = \{x \in P | \alpha^T x = \beta\} = \{A^{=} x = b^{=}, \alpha^T x = \beta, \hat{A} x \leq \hat{b}\}$, where $\alpha^T x \leq \beta$ is an inequality from $A^{\leq} x \leq b^{\leq}$ and $\hat{A} x \leq \hat{b}$ are the remaining inequalities in $A^{\leq} x \leq b^{\leq}$. Then $dim(F) = n - rank \begin{bmatrix} A^{=} \\ \alpha \end{bmatrix}$. Suffice to prove α is not a linear combination of $A^{=}$.
Suppose otherwise there exists λ such that $\alpha = \lambda^T A^{=}$. As $P = \{x | A^{=} x = b^{=}, \alpha^T x \leq \beta, \hat{A} x \leq \hat{b}\}$ is not empty, we have $\beta \geq \lambda^T b^{=}$; i.e., $\{x | A^{=} x = b^{=}\} \subseteq \{x | \alpha^T x = \lambda^T b^{=}\} \subseteq \{x | \alpha^T x \leq \beta\}$, contradicted with the fact that $\alpha^T x \leq \beta$ is not redundant. □

Proposition 3.15. *Let $P = \{x \in \mathbb{R}^n | Ax \leq b\}$ be a nonempty polyhedron. F is a minimal face of P if and only if $\emptyset \neq F \subseteq P$ and F is an affine subsystem, i.e., $F = \{x \in \mathbb{R}^n | A'x = b'\}$ where $A'x \leq b'$ is a subsystem of $Ax \leq b$.*

Proof.

1. If F is a non-empty affine subspace and a face of P, then clearly F cannot have a facet (by Proposition 7.6) and therefore F is a minimal face.

2. Let F be a nonempty minimal face of P. By definition, $F = \{x \in P | A'x = b'\}$ where $A'x \leq b'$ is a subsystem of $Ax \leq b$, and the remaining inequalities are $\hat{A}x \leq \hat{b}$, i.e., $F := \{x | A'x = b', \hat{A}x \leq \hat{b}\}$. W.L.O.G we may remove any inequality from $\hat{A}x \leq \hat{b}$ that is redundant. Since F cannot have any facet, we conclude that $\hat{A}x \leq \hat{b}$ is empty.

□

Finally, we end with a result which is a sophisticated version of Decomposition Theorem 3.4.

Theorem 3.5. *Let $P = \{x \in \mathbb{R}^n | Ax \leq b\}$ be a nonempty polyhedron, described by rational data. Then $P = conv\{v^1, \ldots, v^p\} + cone\{r^1, \ldots, r^q\} +$ lin.hull$\{z^1, \ldots, z^t\}$, where*

1. *P has p minimal faces and v^i is arbirary point from the i^{th} minimal face.*

2. rec.cone(P) *has q mininal proper faces*[2] *and r^i is an arbitrary vector from $F^i \setminus$* lin.space(P)*, where F^i is the i^{th} minimal proper face of* rec.cone(P)

3. *z^i's generate* lin.space(P)*.*

Morover, if A and b are rational , then v's, r's, and z's can be selected to be rational.

3.5 Suggested exercises

1. Consider the polytope that is the convex hull of the points $\{i, i^2\}$ for $i = 0, 1, \ldots, 10$. Describe the facet-defining inequalities of this polytope.

[2] A face of dimension one more that the dimension of the lin.space

2. Given n distinct sets such that $E_1 \subset E_2 \subset E_3 \subset \dots E_n$ in $\mathbb{R}^{100}$, how large can n be if
   (a) each E_i is a linear subspace
   (b) each E_i is a an affine subspace
   (c) each E_i is a convex set
3. Describe the extreme points (minimal faces)and extreme rays (minimal proper faces of recession cone) of the following sets:
   (a) $X = \{x \in \mathbb{R}^n \mid x \geq 0, \sum_{i=1}^n a_i x_i = 1\}$ where $a_1 < a_2 < a_3 < \dots < a_n$.
   (b) $X = \{x \in \mathbb{R}^3 \mid x \geq 0, x_1 + x_2 - x_3 \geq 1\}$
   (c) $X = \{x \in \mathbb{R}^4 \mid x \geq 0, -x_1 = x_2 - 2x_3 \leq 1, -2x_1 - x_3 + 2x_4 \leq 2$
4. Let S^n be the set of $n!$ vectors obtained by permuting the entries of the vector $(1, 2, \dots, n)$ and P^n be the convex hull of S_n. For example for $n = 3$: $S^3 := \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
   (a) Prove that $\dim(P^n) = n - 1$
   (b) Given S, a nonempty, proper subset of $\{1, 2, \dots, n\}$, prove that the inequality
   $$\sum_{j \in S} x_j \geq \frac{|S||S| + 1}{2}$$
   is a facet-defining inequality for P^n.
5. Construct the face-lattice for the polytope $\{x \in \mathbb{R}^3 \mid x \geq 0, x_1 + x_2 + x_3 \leq 1\}$.
6. In class we proved the following: Let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a polyhedron. Let F be a face of P. Then there exists a subsystem $A'x \leq b'$ of $Ax \leq b$ such that $F = \{x \in P \mid A'x = b'\}$. Now prove the converse: Suppose there exists a subsystem $A'x \leq b'$ of $Ax \leq b$ such that $F = \{x \in P \mid A'x = b'\}$ is non-empty. Prove that F is a face of P.
7. In class we proved that if $P \subseteq \mathbb{R}^n$ is a polyhedron and F is a facet of P, then $\dim(P) = \dim(F) + 1$. Now verify the converse: if $P \subseteq \mathbb{R}^n$ is a polyhedron and F is a face such that $\dim(P) = \dim(F) + 1$, then F is a facet.

8. Let P be a non-empty polyhedron. Let F be any face of P. Prove the lin.space(F) = lin.space(P).

9. Let $P \subseteq \mathbb{R}^n$ be a nonempty polytope. Let x^0 be a vertex (0-dimensional minimal face) of P. Let $x^1, \ldots, x^k$ be all the neighboring vertices of x^0, i.e., all the one dimensional faces of P containing x^0 are of the form $\text{conv}\{x^0, x^t\}$ for $t \in \{1, \ldots, k\}$. Prove that if $x \in P$, then there exists $\lambda_t \geq 0$ for $t \in \{1, \ldots, k\}$ such that $x = \sum_{t=1}^{k} \lambda_t (x^t - x^0) + x^0$.

10. Let $P \subseteq \mathbb{R}^n$ be a non-empty polytope. Let vert(P) be the set of vertices of P. Let $X \subseteq \text{vert}(P)$. Define $P(X) := \text{conv}\,(\text{vert}(P) \setminus X)$. The graph of the polytope P is a graph G_P with nodes corresponding to vert(P) such that two nodes are adjacent in G_P if and only if the corresponding vertices are adjacent in P (i.e. the two vertices are contained in a one-dimensional face of P).

    Let $X \subseteq \text{vert}(P)$ and let $(X_1, \ldots, X_m)$ be a partition of X such that X_i and X_j are independent in G_P, i.e. there is no edge connecting X_i to X_j for all $1 \leq i < j \leq m$. Then show that

    $$P(X) = \bigcap_{i=1}^{m} P(X_i).$$

11. Let $P \subseteq \mathbb{R}^n$ be a polytope (bounded polyhedron). Define a vertex of P as follows: if there exists some $c \in \mathbb{R}^n$ such that $c^T v < c^T x$ for all $x \in P \setminus \{v\}$, i.e., it is a 0 dimensional face of P. For a given vertex v, define the set $S^v = \{c \in \mathbb{R}^n \,|\, c^T v < c^T x \text{ for all } x \in P \setminus \{v\}\}$. Prove that the dimension of the set S^v is n.

12. Let $M(c)$ be the set of optimal solutions when maximizing $c^\top x$ over P. The following holds: for every $c \in \mathbb{R}^n$ such that $M(c) \neq P$, $M(c)$ is completely contained in exactly one proper face of Q.

    Prove that $P = Q$.

13. Let $P^i := \{x \,|\; A^i x \leq b^i\}$ $i \in \{1, 2\}$. Suppose that rec.cone(P^1) = rec.cone(P^2). Prove that convex hull of $\cup_{i=1}^{2} P^i$ is given by

    $$\{x \in \mathbb{R}^n \,|\, x = x^1 + x^2,\ A^1 x^1 \leq b^1 \lambda,\ A^2 x^2 \leq b^2 (1 - \lambda),\ 0 \leq \lambda \leq 1\}.$$

Chapter 4

Fundamental Theorem of Integer Programming

4.1 Fundamental Theorem of Integer Programming

Theorem 4.1. *Let $P := \{x \in \mathbb{R}^n | Ax \leq b\}$ be a nonempty rational polyhedron, then:*
(i) conv($P \cap \mathbb{Z}^n$) is a rational polyhedron
(ii) If $P \cap \mathbb{Z}^n \neq \emptyset$, then $rec.cone(conv(P \cap \mathbb{Z}^n)) = rec.cone(P)$

Proof. If $P \cap \mathbb{Z}^n = \emptyset$, the statement trivially holds. Now suppose $P \cap \mathbb{Z}^n \neq \emptyset$.
(1) Since P is a rational polyhedron. By the previous proposition, we have $P = conv(v^1, \ldots, v^p) + cone(r^1, \ldots, r^q)$, where $v^i \in \mathbb{Q}^n$ and $r^j \in \mathbb{Z}^n$.
Since

$$P \supseteq T := \{x \in \mathbb{R}^n | x = \sum \lambda_i v^i + \sum \theta_j r^j, \lambda_i \geq 0, \sum \lambda_i = 1, \theta_j \geq 0, \theta_j \leq 1\}$$

Observe that T is rational and bounded.
For any $\bar{x} \in \mathbb{Z}^n$, define $T_{\bar{x}} = \{x \in T | x = \bar{x}\}$. Then $T \cap \mathbb{Z}^n = \cup_{\bar{x} \in \mathbb{Z}^n} T_{\bar{x}}$. Note that $T_{\bar{x}}$ is a rational polytope (T with one more rational constraint), which can be represented by convex combinations of rational vectors $v^1_{\bar{x}}, \ldots, v^{k_{\bar{x}}}_{\bar{x}}$. It follows that conv($T \cap \mathbb{Z}^n$) can be represented by convex combinations of $v^1_{\bar{x}}, \ldots, v^{k_{\bar{x}}}_{\bar{x}}, \bar{x} \in \mathbb{Z}^n$ and thus is a rational polytope.

We then CLAIM that $P \cap \mathbb{Z}^n = (T \cap \mathbb{Z}^n) + \sum \mu_j r^j$, where $\mu_j \in \mathbb{Z}^n_+$.

Proof. $\supseteq$: by definition
$\subseteq$: for any $x \in P \cap \mathbb{Z}^n$, $x = \sum \lambda_i v^i + \sum \theta_j r^j = \sum \lambda_i v^i + \sum(\theta_j - \lfloor \theta_j \rfloor) r^j +$

$\sum \lfloor \theta_j \rfloor r^j$.
Since $x \in \mathbb{Z}^n$ and $\sum \lfloor \theta_j \rfloor r^j \in \mathbb{Z}^n$, we have $\sum \lambda_i v^i + \sum (\theta_j - \lfloor \theta_j \rfloor) r^j \in \mathbb{Z}^n$. On the other hand, since $0 \leq \theta_j - \lfloor \theta_j \rfloor \leq 1$, we have $\sum \lambda_i v^i + \sum (\theta_j - \lfloor \theta_j \rfloor) r^j \in T$. Thus $\sum \lambda_i v^i + \sum (\theta_j - \lfloor \theta_j \rfloor) r^j \in T \cap \mathbb{Z}^n$.
Note that $\lfloor \theta_j \rfloor \in \mathbb{Z}^n_+$, we conclude that $P \cap \mathbb{Z}^n \subseteq (T \cap \mathbb{Z}^n) + \sum \mu_j r^j$ □

It is well known that for two sets $A, B, \mathrm{conv}(A+B) = \mathrm{conv}(A) + \mathrm{conv}(B)$. It follows that $\mathrm{conv}(P \cap \mathbb{Z}^n) = \mathrm{conv}(T \cap \mathbb{Z}^n) + \mathrm{cone}\{r^1, \ldots, r^q\}$, thus $conv(P \cap \mathbb{Z}^n)$ is a rational polyhedron
(2) If $P \cap \mathbb{Z}^n \neq \emptyset$, from the proof of part (1), we have $\mathrm{rec.cone}(\mathrm{conv}(P \cap \mathbb{Z}^n)) = \mathrm{cone}\{r^1, \ldots, r^q\} = \mathrm{rec.cone}(P)$ □

4.2 Integral Polyhedron

Motivated by the fundamental Theorem, we make the following definition.

Definition 4.1 (Integral Polyhedron)**.** *We say that a polyhedron $P \subseteq \mathbb{R}^n$ is an integral polyhedron if* $\mathrm{conv}(P \cap \mathbb{Z}^n) = P$.

Here is an important result regarding integral polyhedra.

Proposition 4.1. *A rational polyhedron P is integral iff every minimal face contains an integer point.*

Proof. Suppose P is integral. Let F be a minimal face of P, and assume for the sake of obtain an contradiction F contains no integer points. Then $P \setminus F$ is a convex set containing $P \cap \mathbb{Z}^n$, but it is a strict subset of P, which is therefore not the convex hull of $P \cap \mathbb{Z}^n$.

Conversely, if $P^I = \emptyset$, the result holds trivially so assume $P \neq P^I \neq \emptyset$. Suppose, we have $P = \mathrm{conv}\{u^1, \ldots, u^p\} + \mathrm{cone}\{r^1, \ldots, r^k\}$ and $u^1 \neq P^I$. Therefore,

$$(u^1 + \mathrm{lin.space}(P^I)) \cap P^I = \emptyset.$$

By the Fundamental Theorem of Integer programming, the recession cones of P and P^I are the same, and therefore $\mathrm{lin.space}(P^I) = \mathrm{lin.space}(P)$. Thus, we have that $(u^1 + \mathrm{lin.space}(P)) \cap P^I = \emptyset$. However, note that $(u^1 + \mathrm{lin.space}(P))$ is a minimal face of P, completing the proof. □

4.3 A simple example

Let the graph $G = (V, E)$ be a complete graph. Consider the standard formulation of the stable set on this graph.

$$\left.\begin{array}{rcll} x_i + x_j & \leq & 1 & \forall\, (i,j) \in E \\ x_i & \geq & 0 & \forall\, i \in V \end{array}\right\} (P) \tag{4.1}$$

$$x \in \mathbb{Z}^{|V|}. \tag{4.2}$$

Dimension on P^I : Observe that the following points belong to P^I: $(0,\dots,0)$, $(1,0,\dots,0)$, $(0,1,\dots,0)$, $\dots$, $(0,\dots,0,1)$. Clearly, these points $n+1$ are affinely independent. Thus, $\dim(P^I) = n$.

A new valid inequality for P^I: Consider the inequality,

$$\sum_{i=1}^{n} x_i \leq 1. \tag{4.3}$$

This inequality is valid for P^I, since the graph is complete and therefore no more than one vertex can be selected. Observe that this inequality is *not implied* by the inequalities describing P. Thus P is not integral.

The inequality (4.3) is facet-defining for P^I: Examine that the points $(1,0,\dots,0)$, $(0,1,\dots,0)$, $\dots$, $(0,\dots,0,1)$ belong to P^I, satisfy the inequality (4.3) at equality, and are affinely independent. Thus the dimension of the face defined by the inequality (4.3) is $n-1$, i.e. facet-defining. Note that if P^I is not full-dimensional, then more care should be taken in order to prove an inequality is facet-defining, as discussed in class.

Integer hull: Consider the set:

$$Q = \{x \in \mathbb{R}^n \mid x_i \geq 0 \ \forall i \in [n], \sum_{i=1}^{n} x_i \leq 1\}.$$

We claim that $Q = P^I$.

Proof. Observe first that $Q \cap \mathbb{Z}^n = P \cap \mathbb{Z}^n$. Therefore to complete the proof we have to show that Q is integral. Note now that the extreme points (minimal faces) of Q are $(0,\dots,0)$, $(1,0,\dots,0)$, $(0,1,\dots,0)$, $\dots$, $(0,\dots,0,1)$ and thus integral. Therefore, by Proposition 4.1, Q is integral. $\square$

4.4 Proving a Polyhedron is integral

Clearly, given an IP, we would like to obtain its integral hull, since optimizing a linear function on the integer hull is a linear program (thanks to the Fundamental Theorem) and it gives you the exact same optimal objective function value as the original IP. Unfortunately, in most cases, finding the integer hull is challenging. However in special cases, we are ale to prove that we have the exact integer hull. Proving such a result usually involves proving that some polyhedron is integral. Many different methods have been used to prove this. Here are some examples of techniques for showing a rational polyhedron P is integral (or $P = \text{conv}(X)$ where $X = P \cap \mathbb{Z}^n$):

1. Show that all extreme points of P are integral. The contrapositive is:

2. Show that all points of P with $x \notin \mathbb{Z}^n$ are not extreme points of P.

3. Show that all facets/faces of P have integral extreme points.

4. Show that the linear program: $\min\{c^\top x \,|\, x \in P\}$ has an optimal solution in X for all $c \in \mathbb{R}^n$. One way to do this is:

5. Show that there exists a point $x^* \in X$ and point u feasible in the dual of the linear programming relaxation with $c^\top x^* = (u)^\top b$. (Why does this work?) Another is:

6. Show that the matrix corresponding to the left-hand-side of constraint defining P is totally unimodular and the right-hand-ide is an integral vector. See detail in Chapter 5.

7. Assuming $b \in \mathbb{Z}^m$, show that, for all $c \in \mathbb{Z}^n$ for which the optimal dual objectuve function is bounded, it is also integer valued. This is related to the Totally dual integral polyhedron. See Chapter 5.

8. Show that $\dim(\text{conv}(X)) = \dim(P)$, and that if $\pi^\top x \geq \pi^0$ is a facet-defining inequality for conv(X), then $\pi^\top x \geq \pi^0$ must be identical to one of the inequalities defining P.

9. Show that $\dim(\text{conv}(X)) = \dim(P)$, and that for all $c \in \mathbb{R}^n$ for which $M(c) \neq X$ and the optimum value is finite, $M(c) \subseteq \{x \,|\, \alpha^\top c = \beta\}$ for some inequality $\alpha^\top x \geq \beta$ defining P, where $P \cap \{x \,|\, \alpha^\top c = \beta\}$ is a proper face of P. Here $M(c) = \arg\min\{c^\top x : x \in X\}$ is the set of optimal solutions for a given cost vector $c \in \mathbb{R}^n$. (See Problem 11 in Chapter 3).

10. Another possibility is to show Q is a mixed integer set such that (1) $X = \text{proj}_x(Q)$, (2) conv(Q) is integral and $\text{proj}_x(\text{conv}(Q)) = P$.

4.5 PORTA

PORTA is an useful code to determine facet-defining inequalities experimentally.

4.5.1 Downloads

For Windows and Linux users Porta is available at "https://porta.zib.de/". Additionally the page contains links to other useful tools, such at cdd and Normalize.

For Mac users the easiest way is likely to just compile the C code on your machine. The source code for various versions are available on online, one such can be found at this github link "https://github.com/denisrosset/porta".

4.5.2 File Types

Porta makes use of two basic file types, the ".ieq" file and the ".poi" files. The ".ieq" file type corresponds to the outer description of a polyhedron and ".poi" files correspond to the inner description. While some functions in Porta require a specific file type for input (fmel for example requires an ".ieq" file), but the primary function traf accepts either file type.

.ieq file

The ".ieq" file contains the outer description of the polyhedron. The basic structure is as follows.
****
DIM = n (where n is the number of variables in your problem)
VALID
(a vector corresponding to a feasible point)
INEQUALITIES_SECTION
(your constraints)
END

************************* Inequalities in Porta are written in the form a_{i1}x1 +/- a_{i2}x2 .... +/- a_{in}xn (==, <=, >=) b_i, where the a_{ij} and b_i are the data for your problem. Note that the data for your problem must be rational.

.poi file

The ".poi" file contains the inner description of your polyhedron. The basic structure is as follows.
****
DIM = n (where n is the number of variables in your problem)
CONV_SECTION
(rational vectors corresponding to your polyhedron's extreme points)
CONE_SECTION
(rational vectors corresponding to your polyhedron's extreme rays)
END

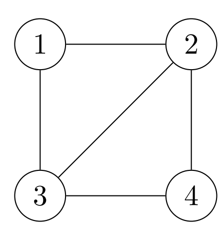
****

4.5.3 traf

The primary function in Porta is traf. The traf function converts from a polyhedra's outer description to its inner description and its inner description to its outer description. In Porta terms, traf converts an ".ieq" file to its corresponding ".poi" file and a ".poi" file to its correspdonding ".ieq" file. To use traf, you must call the traf program and provide it an ".ieq" or ".poi" file as an argument. For example, running traf from a linux command line looks like, "./traf file.(ieq/poi)". After being ran, traf outputs a new file named after your input file with the appropriate extension appended to the end. For example, if you ran traf on a file called "file.ieq" traf would output a file called "file.ieq.poi".

Example

Consider the stable set problem on the graph below.



The extreme points we are interested in are

$$(0, 0, 0, 0), (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1), (1, 0, 0, 1)$$

The ".poi" file representing our polytope looks like

```
DIM = 4
CONV_SECTION
0 0 0 0
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
1 0 0 1
END
```

Calling traf on this file will then output the following ".ieq" file containing the outer description of our integral stable set polytope.

```
DIM = 4
VALID
1 0 0 1
INEQUALITIES_SECTION
-x1 <= 0
-x2 <= 0
-x3 <= 0
-x4 <= 0
+x2+x3+x4 <= 1
+x1+x2+x3 <= 1
END
```

4.5.4 fmel

Porta essentially works based on an implementation of Fourier-Motzkin projection. The fmel function can be used to project polyhedra into a lower dimensional spaces. To do this you use the fmel function with an ".ieq" file as an input. The ".ieq" file must contain an additional line denoting the order in which variables are to be eliminated during the Fourier-Motzkin procedure. For example, suppose you had a polyhedra with four variables, if you wanted to eliminate the second and third of these you could add the following line to your ".ieq" file.

ELIMINATION_ORDER
0 1 2 0

Running fmel on the file would then produce a new ".ieq" file without the variables you selected for elimination.

4.6 Suggested exercises

1. Let m and n be positive integers. Let $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$ and $A \in \mathbb{Q}^{m\times n}$. Suppose that the LP $\max\{c^T x \mid Ax \leq b\}$ is unbounded. Which of the following outcomes of the IP problem $\max\{c^T x \mid Ax \leq b, x \in \mathbb{Z}^n\}$ are possible: unbounded, infeasible, have optimal solutions. Provide arguments for your answer.

2. Let $n \geq 3$ and $m \geq 2$ be two integers. The simple plant location polytope is the convex hull of the points

$$\begin{aligned} \sum_{j=1}^{n} x_{ij} &= 1 \ \forall i = 1, \ldots, m \\ 0 \leq x_{ij} \leq y_j &\leq 1 \ \forall i = 1, \ldots, m, j = 1, \ldots, n \\ x_{ij} \in \{0,1\},\ y_j &\in \{0,1\} \forall i = 1, \ldots, m, j = 1, \ldots, n. \end{aligned}$$

   (a) Find the dimension of the simple plant location polytope.

   (b) Show that $x_{ij} \geq 0$ defines a facet of the simple plant location polytope.

3. A set S is a called mixed integer representable if there exists rational matrices A, B, C and rational vector d such that

$$S = \{x \in \mathbb{R}^n \mid Ax + By + Cz \leq d, y \in \mathbb{R}^p, z \in \mathbb{Z}^q\}.$$

   Prove that a set S is mixed integer representable iff there exists rational polytopes $P_1, \ldots, P_k \in \mathbb{R}^n$ and integral vectors $r^1, \ldots, r^t$ such that

$$S = \bigcup_{i=1}^{k} P_i + \left\{ \sum_{i=1}^{t} r^i u_i \mid u_i \in \mathbb{Z}_+ \right\}.$$

4. Prove the Fundamental theorem of integer programming for the case of mixed integer linear program, i.e. if $P \subseteq \mathbb{R}^n$ is a rational polyedron then

   - $\text{conv}(P \cap (\mathbb{Z}^{n_1} \times \mathbb{R}^{n_2})$ is a polyhedron where $n_1 + n_2 = n$.
   - If $\text{conv}(P \cap (\mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}) \neq \emptyset$,

   $$\text{rec.cone}\,(\text{conv}(P \cap (\mathbb{Z}^{n_1} \times \mathbb{R}^{n_2})) = \text{rec.cone}(P).$$

5. Let $(a_1, \ldots, a_n) \in \mathbb{Z}^n_+ \setminus \{0\}, b \in \mathbb{R}$ and $S = \{x \in \mathbb{Z}^n \mid \sum_{j=1}^n a_j x_j \leq b\}$. Note that x is allowed to be general integer and not restricted to only non-negative integers. Prove that

   $$\text{conv}(S) = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{j=1}^n \frac{a_j}{k} x_j \leq \left\lfloor \frac{b}{k} \right\rfloor \right\},$$

   where k is the greatest common divisor of $a_1, \ldots, a_n$.

6. Let $S = \{x \in \{0,\ 1\}^n, y \in \mathbb{R}_+ \,|\, y + a_i x_i \geq a_i \ \forall i \in \{1, \ldots, n\}\}$ where $a_1 > a_2 > \cdots > a_n > 0$. Prove that

   $$y + (a_1 - a_2)x_1 + (a_2 - a_3)x_2 + \cdots + (a_{n-1} - a_n)x_{n-1} + a_n x_n \geq a_1$$

   is valid and facet-defining for $\text{conv}(S)$.

7. Consider the set $S = \{(x, y) \in \mathbb{R}^n \times \mathbb{Z} : \sum_{j=1}^n x_j \leq ny, 0 \leq x_j \leq 1 \ \forall j, 0 \leq y \leq 1\}$. Prove that $\text{conv}(S) = \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : 0 \leq x_j \leq y \ \forall j,\ y \leq 1\}$.

8. We have n binary variables x_i for $i \in [n]$, and we are interested in all products of distinct binary variables, $y_{ij} = x_i x_j$ where $i \neq j$. Then it is easy to 'linearize' this product as

   $$y_{ij} \geq 0 \ \forall i, j \in [n], i \neq j \tag{4.4}$$
   $$y_{ij} \leq x_i \ \forall i, j \in [n], i \neq j \tag{4.5}$$
   $$y_{ij} \leq x_j \ \forall i, j \in [n], i \neq j \tag{4.6}$$
   $$y_{ij} \geq x_i + x_j - 1 \ \forall i, j \in [n], i \neq j \tag{4.7}$$
   $$x \in \mathbb{Z}^n, y \in \mathbb{Z}^{n(n-1)/2}. \tag{4.8}$$

   Thus we arrive at the following IP set: S:=

   $$\left\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^{n(n-1)/2} \,|\, (x, y) \text{ satisfies } (4.4), (4.5), (4.6), (4.7), (4.8)\right\}$$

- Show that (4.4), (4.5), (4.6), (4.7) imply that $0 \leq x_i \leq 1$ for all $i \in [n]$, and $y_{ij} \leq 1 \; \forall i, j \in [n], i \neq j$.
- What is the dimension of conv(S)?
- Show that $y_{ij} \leq x_i$ is a facet-defining inequality of conv(S) $\forall i, j \in [n], i \neq j$.
- Show that for $n = 2$, conv(S) is given by the inequalities (4.4), (4.5), (4.6), (4.7).
- Show that for $n = 3$, the following inequality is valid:

$$x_1 + x_2 + x_3 - y_{12} - y_{13} - y_{23} \leq 1.$$

- Prove that above inequality is a facet-defining inequality for the case of $n = 3$.

9. Given an undirected graph $G = (V, E)$ with n vertices, $V = \{1, \ldots, n\}$, a dominating set in G is a set of vertices D such that every vertex in the graph is either in D or is adjacent to some vertex in D, i.e. for every vertex $i \in V$ either $i \in D$ or $(i, j) \in E$ for some $j \in D$. The dominating set polytope is conv(Q) where

$$Q = \{x \in \{0,1\}^n \mid \sum_{j \in \Delta(i)} x_j \geq 1 \; \forall i \in V\},$$

and where $\Delta(i) = \{i\} \cup \{j \in V \mid (i, j) \in E\}$ is the set of vertices adjacent to vertex i together with i itself.

(a) (15) Assume that G contains no isolated edges. Prove that conv(Q) is full-dimensional.

(b) (15) Suppose that G is a wheel graph, i.e., $G = (V, E)$ where $V = \{1, \ldots, n, n+1\}$ (where $n \geq 4$) and

$$E = \{(i, i+1) \mid i \in \{1, \ldots, n-1\}\} \cup (n, 1) \cup \{(i, n+1) \mid i \in \{1, \ldots, n\}\}.$$

Show that the following is a valid inequality for the dominating set polytope for the wheel graph:

$$\sum_{i=1}^{n} x_i + \left\lceil \frac{n}{3} \right\rceil x_{n+1} \geq \left\lceil \frac{n}{3} \right\rceil$$

(c) (15) Show that above inequality is facet-defining for the dominating set polytope for the wheel graph.

10. Let $G = (V, E)$ be a simple graph. Remember that a stable set is a subset V' of the vertices V such that for all $u, v \in V'$, $(u, v) \notin E$. Therefore the set of all stable sets may be represented as:

$$\begin{aligned} &x_u + x_v \le 1 \ \forall (u,v) \in E \\ &0 \le x_u \le 1 \ \forall u \in V \\ &x \in \mathbb{Z}^{|V|}. \end{aligned} \tag{4.9}$$

   (a) Consider the graph C_5 (cycle on 5 vertices), i.e. $V = \{v_1, v_2, v_3, v_4, v_5\}$ and
   $$E = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_1)\}.$$
   Show that LP relaxation of (4.9) for the graph C_5 is not integral.

   (b) Prove that the inequality
   $$\sum_{i=1}^{5} x_i \le 2, \tag{4.10}$$
   is a facet-defining inequality of the convex hull of (4.9) for the graph C_5.

11. In production of electricity, the resources are generators which produce electricity in various time periods and we are planning for T time periods. A generator has a minimum up time of U i.e., if a generator is turned on in time period t, then it must remain on for the next $U - 1$ time periods. One may model such constraints as:

$$\begin{aligned} x_t - x_{t-1} \le x_\tau \text{ for all } 2 \le t < \tau \le \min\{t + U - 1, T\} \\ x_t \in \{0, 1\} \text{ for all } 1 \le t \le T \end{aligned}$$

We are interested polyhedral description of the convex hull of the above set (call this set S). For a nonnegative integer k, consider a non-empty set of $2k + 1$ indices (denoted as $\{\phi(1), \phi(2), \ldots, \phi(k + 1), \psi(1), \psi(2), \ldots, \psi(k))$ from the set $\{1. \ldots, T\}$ such that

   (a) $\phi(1) < \psi(1) < \phi(2) < \psi(2) < \cdots < \phi(k) < \psi(k) < \phi(k + 1)$
   (b) $\phi(k + 1) - \phi(1) \le U$

Associate with these indices the following inequalities:

$$-\sum_{j=1}^{k+1} x_{\sigma(j)} + \sum_{j=1}^{k} x_{\psi(j)} \le 0$$

Prove that these inequalities are valid and are facet-defining for the convex hull of S.

12. Let $T \subseteq \{0,1\}^n$. We say that T satisfies the *edge property* if for all $c \in \mathbb{R}^n$ such that $\min\{c^t z \mid z \in T\}$ has at least two optimal solutions, z^1 and z^2 where $\sum_{j=1}^n z^1_j = k^1$, $\sum_{j=1}^n z^2_j = k^2$ and $k^1 \leq k^2 - 2$, then there is an optimal solution z^3 such that $\sum_{j=1}^n z^3_j = k^3$ and $k^1 < k^3 < k^2$.
    (a) If T satisfies the edge property, prove that
    $$\mathrm{conv}\left(T \cap \{z \in \{0,1\}^n \mid \sum_{j=1}^n z_j \leq k\}\right) = \mathrm{conv}(T) \cap \{z \in [0,1]^n \mid \sum_{j=1}^n z_j \leq k\}$$
    for any $k \in \{1, \dots, n\}$.
    (b) Prove that the set of matchings in a graph $G = (V,E)$, i.e., $\{x \in \{0,1\}^{|E|} \mid \sum_{e \in \delta(v)} x_e \leq 1 \forall e \in E\}$ satisfies the edge property.

13. (a) Let $P := \{(x,y) \in [0,1]^p \times \mathbb{R}^q \mid Ax + By \leq d\}$ be a non-empty integral polytope, that is for any vertex of P, it's x-component is a $0-1$ vector and it's y-component is a vector of general integers. Let $l \leq p$. Suppose there is a vertex of P such the first l components of x is the vector $\bar{x} \in \{0,1\}^l$. Prove that the polytope Q is integral where Q is defined as:
    $$Q = P \cap \{(x,y) \in [0,1]^p \times \mathbb{R}^q \mid x_j = \bar{x}_j \ \forall j \in \{1, \dots, l\}\}.$$
    (b) Would the same result hold if x-component of P is general integer instead of being $0-1$ vector?

14. We will show that the decision version of IP (Given $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$, is the set $\{x | Ax \leq b, x \in \mathbb{Z}^n\}$ non-empty?) is in NP. For simplicity assume that $Ax \leq b$ is pointed. The certificate will be a feasible solution.

    Hint: You can re-use the proof of the fundamental theorem of IP. Remember the set $T = \{x | x = \mathrm{conv}\{v^1, \dots, v^p\} + \sum_{j=1}^q \theta_j r^j, 1 \geq \theta_j \geq 0, \forall j \in \{1, \dots, q\}\}$ where the polyhedron $P := \{x | Ax \leq b\} = \mathrm{conv}\{v^1, \dots, v^p\} + \mathrm{cone}\{r^1, \dots, r^q\}$ where r^j are scaled to be integral. Our proof of the fundamental theorem already shows that $\{x | Ax \leq b, x \in \mathbb{Z}^n\}$ is non-empty iff $T \cap \mathbb{Z}^n$ is non-empty. All you need to show is the size of the integer feasible solutions in T is bounded by a polynomial function of the size of the input. Assume/use the solution of Problem 1 in HW 2.

Chapter 5

Theory of Totally Unimodular Matrices and Totally Dual Integral Systems

5.1 Totally Unimodular Matrices and Their Applications in IP

Definition 5.1 (Totally Unimodular (TU)). *An $m \times n$ integral matrix A is called* ***totally unimodular*** *if every square sub-matrix of A has determinant $+1$, -1 or 0.*

Theorem 5.1. *If a matrix A is TU, then a non-empty polyhedron $P := \{x|Ax \leq b\}$ is integral for all integral b.*

Proof. In the view of Proposition 4.1, we only need to show that, for any integral b, every minimal face of P contains an integral point. Recall that any minimal face F of P has a representation

$$F := \{x|A'x = b'\},$$

where $\{A'x = b'\}$ is a subsystem of $\{Ax = b\}$. Without loss of generality, we can assume that the matrix A' has linearly independent rows. We denote $A = [B, \bar{B}]$, where B is a full rank (hence invertible) square matrix. Rewrite $A'x = b'$ as

$$Bx_B + \bar{B}x_{\bar{B}} = b,$$

where $x = (x_B, x_{\bar{B}})$. Taking $x_{\bar{B}} = 0$, we have $x = (B^{-1}b, 0)$ as a point contained in the face F. Since A is TU, we know that $\det(B^{-1}) = \pm 1$. Given that b is integral, $B^{-1}b$ is integral and thus x is integral. Thus, we have established that any face F of P contains an integral point. □

Is it true that whenever we have an integral polyhedral, the left-hand-side is TU? This is not true as the next example shows

Example 5.1. *Consider the polyhedron* $P = \{x \in \mathbb{R}^3 \mid x_1 + x_2 \leq 2, x_1 + x_3 \leq 2, x_2 + x_3 \leq 2, x \geq 0\}$*. The extreme points of this polyhedron are:* $(0,0,0)$*,* $(2,0,0)$*,* $(0,2,0)$*,* $(0,0,2)$*, and* $(1,1,1)$*, therefore the polytope is integral. On the other hand the constrint matrix is not TU, since*

$$\dim\left(\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}\right) = 2.$$

5.1.1 Hoffman and Kruskal Theorem

Although integral polyhedron doen not imply the left-hand-side is TU, the following result, due to Hoffman and Kruskal, is true.

Theorem 5.2. *Let A be an integral matrix, then $P(b) = \{x \mid Ax \leq b, x \geq 0\}$ is integral for all integral b (when $P(b) \neq \emptyset$) if and only if A is TU.*

Before giving the proof of Theorem 5.2, let us first introduce the concept of *"unimodular matrix"* and several useful lemmas.

Definition 5.2 (Unimodular)**.** *A $m \times n$ matrix A is called unimodular if*

1. *A is integral and has full row rank.*
2. *Every $m \times m$ square submatrix of A has determinant $+1$, -1 or 0.*

Lemma 5.1. *Let B be an integral invertible square matrix. If $B^{-1}t$ is integral for all integral vector t, then $|\det(B)| = 1$.*

Proof. Let t be the i-th elementary vector e^i, i.e., all the components are zero except that the i-th component is equal to 1. We know that $B^{-1}e^i$ results in the i-th column of B^{-1}. Therefore, the i-th column of B^{-1} is integral, and thus the matrix B^{-1} is integral. Since computing the determinant of a matrix only involves "addition" and "multiplication", the two arithmetic operations, we can see that $\det(B)$ and $\det(B^{-1})$ are integral. Since

$$\det(B) \cdot \det(B^{-1}) = 1,$$

we have $\det(B) = \pm 1$ and $|\det(B)| = 1$. □

Lemma 5.2. *Let A be an integral matrix with full row rank, then $Q(b) := \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ is integral for all integral b if and only if A is unimodular.*

Proof. Assume A is $m \times n$ matrix. "if": trivial. In fact, for each minimal face of P is of the form $Ax = b, x_i = 0,\ \forall i \in N$ (for some $N \subseteq [n]$ where $|N| = n - m$) contains an integral point given that the face is nonempty.

"only if": Let B be a non-singular $m \times m$ square sub-matrix of A. In view of Lemma 5.1, we only need to show that $B^{-1}t$ is integral for all integral t. Let $t \in \mathbb{Z}^m$, we can take some $y \in \mathbb{Z}^m_+$ such that

$$z := y + B^{-1}t \geq 0.$$

We further let

$$b := bz = B(y + B^{-1}t) = By + t.$$

Note that both By and t are integral, so we know that B is also integral. Furthermore, since z is nonnegative, we know that by construction $(z, 0) = (B^{-1}b, 0)$ is a vertex in P. Given that $Q(b)$ is integral, the vertex $(z, 0)$ is integral. Hence, $z = y + B^{-1}t$ is integral. Since y is integral, we know $B^{-1}t$ is integral. □

Lemma 5.3. *A matrix $\mathbf{A} \in \mathbb{Z}^{m\times n}$ is totally unimodular if and only if $[A, I]$ is unimodular.*

Proof. "⇒" Let $\mathbf{A}$ be a TU matrix. Notice that $[A, I]$ is clearly full row rank and integral. Let's consider a $m \times m$ submatrix $\mathbf{B}$ of $[A, I]$. If all columns belong to $\mathbf{A}$, by the total unimodularity $\mathbf{A}$, we know $det(\mathbf{B}) \in \{0, \pm 1\}$. If there is at least one column in $\mathbf{B}$ belonging to $\mathbf{I}$, say $\mathbf{e}_i$, then $|det(\mathbf{B})| = |det(\mathbf{B}')|$, where $\mathbf{B}'$ is obtained by removing the column of $\mathbf{e}_i$ and ith row. We can continuously get a smaller submatrix until it is a submatrix of $\mathbf{A}$. By the total unimodularity of $\mathbf{A}$, we know $det(\mathbf{B}) \in \{0, \pm 1\}$. Therefore, $[A, I]$ is unimodular.

"⇐" Pick any submatrix $\mathbf{A}'$ of $\mathbf{A}$, with row indices $\{i_1, ..., i_t\} \subseteq \{1, ..., m\}$ and column indices $\{j_1, ..., j_t\} \subseteq \{1, ..., n\}$. Let $\{i_{t+1}, ..., i_m\} = \{1, ..., m\} \setminus \{i_1, ..., i_t\}$. Consider a matrix $Y = [\mathbf{A}_{j_1}, ..., \mathbf{A}_{j_t}, \mathbf{e}_{i_{t+1}}, ..., \mathbf{e}_{i_m}]$, this is an $m \times m$ submatrix of $[A, I]$. By definition of determinant, we have $|det(\mathbf{A}')| = |det(\mathbf{Y})| \in \{0, 1\}$. Therefore, $\mathbf{A}$ is totally unimodular. □

Proof. of Theorem 5.2 First, by Lemma and Proposition above, we have "$\mathbf{A}$ is totally unimodular" ⇔ "$[A, I]$ is unimodular" ⇔ "$Q(\mathbf{b}) = \{\mathbf{x} \in \mathbb{R}^n_+, \mathbf{y} \in$

$\mathbb{R}^m_+ : \mathbf{Ax} + \mathbf{y} = \mathbf{b}\}$ is integral for all integral vectors $\mathbf{b} \in \mathbb{Z}^m$ for which $Q(\mathbf{b}) \neq \emptyset$."

Next, we only need to prove "$Q(\mathbf{b})$ is integral" $\Leftrightarrow$ "$P(\mathbf{b})$ is integral". If $Q(\mathbf{b})$ is integral, for any extreme point $\mathbf{x}$ in $P(\mathbf{b})$, $(\mathbf{x}, \mathbf{b} - \mathbf{Ax})$ is an extreme point of $Q(\mathbf{b})$, and thus $\mathbf{x}$ is integral. Then we know $P(\mathbf{b})$ is integral. If $P(\mathbf{b})$ is integral, for any extreme point $(\mathbf{x}, \mathbf{y})$ in $Q(\mathbf{b})$, $\mathbf{x}$ is an extreme point of $P(\mathbf{b})$, and thus $(\mathbf{x}, \mathbf{y})$ is integral. (Since $\mathbf{x}$ is active at n linearly independent constraints in $\mathbf{Ax} \leq \mathbf{b}$.) Therefore, we complete the proof. $\square$

5.1.2 Testing for TU

It turns out that given a mtrix it can be checked in polynomial time whether a matrix is TU or not. We do not present this result here. What we present is Theorem 5.3 which is quite useful to verify TU property, although it does not directly lead to a polynomial-time algorithm. We begin with a preliminary Lemma.

Lemma 5.4. *(i) A matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is TU iff $\begin{bmatrix} \mathbf{A} \\ \mathbf{I} \end{bmatrix}$ is TU.*

(ii) A matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is TU iff $\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix}$ is TU.

(iii) A matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is TU iff $\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \\ \mathbf{I} \end{bmatrix}$ is TU.

(iv) A matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is TU iff $\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \\ \mathbf{I} \\ -\mathbf{I} \end{bmatrix}$ is TU.

Proof. (i) Similar to Lemma 1.

(ii) If $\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix}$ is TU, it is clear that $\mathbf{A}$ is TU. On the other hand, if $\mathbf{A}$ is TU, let's consider a submatrix $\mathbf{A}'$ of $\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix}$. If $\mathbf{A}'$ contains the same row from $\mathbf{A}$ and $-\mathbf{A}$, then $det(\mathbf{A}') = 0$. Otherwise $\mathbf{A}'$ is the same as $\mathbf{A}$ with row replacement. And thus $|det(\mathbf{A}')| = |det(\mathbf{A})|$. Therefore, $\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix}$ is TU.

(iii) This follows from (i) and (ii).

(iv) This follows from (i), (ii) and (iii). $\square$

Theorem 5.3. *A matrix $\mathbf{A} \in \mathbb{Z}^{m\times n}$ is totally unimodular if and only if for every $J \subseteq \{1,\ldots,n\}$ there is a partition $J = J_1 \cup J_2, (J_1 \cap J_2 = \emptyset)$ such that,*

$$\sum_{j\in J_1} A_{ij} - \sum_{j\in J_2} A_{ij} \in \{-1,0,1\}$$

for all $i = 1,...,m$. Note that J_i can be empty.

Proof. "$\Rightarrow$" Let $\mathbf{A}$ be totally unimodular and consider a collection J of columns of $\mathbf{A}$. Let be the characteristic vector of J, i.e., $d_j = 1$ when $j \in J$, $d_j = 0$ otherwise. Consider the polytope,

$$P = \{x \in \mathbb{R}^n \,|\, \mathbf{0} \le \mathbf{x} \le \mathbf{d}, \left\lfloor \frac{1}{2}\mathbf{Ad} \right\rfloor \le \mathbf{Ax} \le \left\lceil \frac{1}{2}\mathbf{Ad} \right\rceil\}.$$

It is clear that P is nonempty (since $\mathbf{d}/2 \in P$) and bounded, thus there exists at least one extreme point, $\mathbf{x}_0$. By Lemma 2 and Theorem 1, we know P is integral, thus $\mathbf{x}_0 \in \{0,1\}^n$. Let $\mathbf{y} = \mathbf{d} - 2\mathbf{x}$. If $d_i = 1$, then $d_i - 2x_i$ is either -1 or 1. If $d_i = 0$, then $d_i - 2x_i = 0$ (since $x_i \le d_i$). Therefore, $\mathbf{y}$ defines a partition of columns in J, i.e., $J_1 = \{j : y_j = 1\}$ and $J_2 = \{j : y_j = -1\}$. We show next that $(\mathbf{Ay})_i \in \{0,\pm1\}$ for all $i = 1,...,m$. If $(\mathbf{Ad})_i = 2k+1$, then $k \le (\mathbf{Ax})_i \le k+1$. Thus

$$(\mathbf{Ay})_i = (\mathbf{Ad})_i - 2(\mathbf{Ax})_i = \begin{cases} 1, & (\mathbf{Ax})_i = k, \\ -1, & (\mathbf{Ax})_i = k+1, \end{cases}$$

If $(\mathbf{Ad})_i = 2k$, then it is clear that $(\mathbf{Ay})_i = 0$. Therefore, we have proven that for all $i = 1,...,m$,

$$\sum_{j\in J_1} A_{ij} - \sum_{j\in J_2} A_{ij} \in \{-1,0,1\}$$

"$\Leftarrow$" The proof will be performed by induction on the size of square submatrices of A.
Start with the base case and consider $|J = \{j\}| = 1$. $\sum_{j\in J_1} A_{ij} - \sum_{j\in J_2} A_{ij} \in \{-1,0,1\}$ yields $A_{ij} \in \{0,\pm1\}$.
Now assume for every $(k-1)\times(k-1)$ submatrix A' we have that $det(\mathbf{A}') \in$

$\{0, \pm 1\}$. Now consider a $k \times k$ submatrix $\mathbf{B}$. Using Cramer's rule, we have that

$$(B^{-1})_{ij} = \frac{\det(B^1, B^2, \ldots, B^{i-1} e_j B^{i+1}, \ldots B^k)}{\det(B)},$$

where B^l is the l^{th} column of B and e_j is the unit vector with zeros in every component except j. Thus we have that

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

where $B^* \in \{-1, 0, +1\}^{k \times k}$. Let y be the forst column of B^*. Then we have $\mathbf{By} = [det(\mathbf{B}), 0, ..., 0]^\top$. Let $J := \{j | y_j \neq 0\}$.

Claim 1. $|j \in J | B_{ij} \neq 0|$ is even for all $i \geq 2$.
Proof $\sum_{j \in J} B_{ij} y_j = 0, \forall i \geq 2$ The only possibility that we get 0 is to have the same number of 1's and -1's. Thus $|j \in J | B_{ij} \neq 0|$ is even for all $i \geq 2$.

According to the assumption, there exists a partition J_1, J_2 of J, such that $\sum_{j \in J_1} B_{ij} - \sum_{j \in J_2} B_{ij} \in \{-1, 0, 1\}$. By Claim 1, for $i \geq 2$, we further have $\sum_{j \in J_1} B_{ij} - \sum_{j \in J_2} B_{ij} = 0$. Let's consider $\mathbf{z} \in \{0, \pm 1\}^k$, such that

$$z_j = \begin{cases} 0, & j \notin J, \\ 1, & j \in J_1, \\ -1, & j \in J_2, \end{cases}$$

Observe that $\mathbf{Bz} \neq 0$, since $\mathbf{z} \neq 0$ and $\mathbf{B}$ is non-singular. So we have $\sum_{j \in J_1} B_{ij} - \sum_{j \in J_2} B_{ij} = \pm 1$ for $i = 1$, i.e., $\mathbf{Bz} = [\pm 1, 0, ..., 0]^\top$. Because $\mathbf{B}$ is invertible, comparing $\mathbf{By} = [det(\mathbf{B}), 0, ..., 0]^\top$, we know $\mathbf{y}$ is a scaling of $\mathbf{z}$. On the other hand, we know $\mathbf{y}, \mathbf{z} \in \{0, \pm 1\}^k$. So we have $|det(\mathbf{B})| = 1$ □

This property is based on columns; one can also prove such a statement with rows.

An **interval matrix** is a 0-1 matrix where all the 1's appear in consecutive columns in each row. Then for any collection J of columns, let J_1 be the odd columns, and J_2 be the even columns, it is clear that the characterization in Theorem 2 is satisfied. Hence, any interval matrix is TU.

The **node-edge incidence matrix** of a bipartite graph is also totally unimodular. This follows from the above characterization of TU matrices: Let $\mathbf{A}$ be the node-edge incidence matrix of a bipartite graph $G(V_1; V_2)$. Given an subset of rows I, partition the rows into two set I_1 and I_2 where

I_1 corresponds to nodes in V_1 and I_2 corresponds to nodes in V_2. Then for column j, we have that $\sum_{j\in I_1} A_{ij} \leq 1$ and $\sum_{j\in I_2} A_{ij} \leq 1$ completing the proof.

5.2 Totally Dual Integral

We begin with some preliminary results.

Theorem 5.4 (Integer Farkas Lemma)**.** *Let $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$. Then $\{x \mid Ax = b\} \cap \mathbb{Z}^n = \emptyset$ iff there exists $y \in \mathbb{Q}^m$ such that $y^\top A \in \mathbb{Z}^{1\times n}$ and $y^\top b \notin \mathbb{Z}$.*

Proposition 5.1. *Let $\sum a_i x_i = b_i$, a_i and b_i are integers, and gcd(a_i)=1. Then there exists $\hat{x} \in \mathbb{Z}^n$ such that $a^\top \hat{x} = b$.*

Proposition 5.2. *Let $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a rational polyhedron. Then P is integral, if every rational supporting hyperplane of P contains integer points.*

Proof. P is integral $\rightarrow$ every face of P contains an integer point, which implies that every rational supporting hyperplane of P contains an integer point.

For the other direction, assume by contradiction that every rational supporting hyperplane of P contains an integer point, but P is not integral. Then there exists a minimal face $F := \{x \mid A'x = b'\}$ such that $F \cap \mathbb{Z}^n = \emptyset$ where $A'x \leq b'$ is a subsystem of $Ax \leq b$. By Theorem 5.4 this implies there exists $y \in \mathbb{Q}^{m'}$ such that

$$y^\top A' \in \mathbb{Z}^n, y^\top b' \notin \mathbb{Z}.$$

For any such y, we can find $z \in \mathbb{Z}^{m'}$ such that

$$\begin{aligned} w &:= y + z \geq 0, \\ c &:= w^\top A' \in \mathbb{Z}^n, \\ \delta &:= w^\top b' \notin \mathbb{Z}. \end{aligned}$$

Now consider the equality $c^\top x = \delta$. It is a (rational) supporting hyperplane of P since $A'x \leq b'$ $\forall x \in P$, $w \geq 0$, this implies $c^\top x \leq \delta$ is a valid inequality for P. Moreover, $F \neq \emptyset$, implies $\{x \mid c^\top x = \delta\} \cap P = \emptyset$. However, it does not contain an integer point since $c \in \mathbb{Z}^n$ and $\delta \notin \mathbb{Z}$. This is the required contradiction. $\square$

Proposition 5.3. *Let $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a rational polyhedron. Define $z(c) = \max c^\top x$ such that $x \in P$. Then for any $c \in \mathbb{Z}^n$ such that $z(c) < \infty$, $z(c)$ is integral if and only if P is integral.*

Proof. ($\Leftarrow$) If P is integral, every minimal face contains an integer point. Since the set of optimal solution contains a minimal face, optimization of P will yield in integer objective function value. And for integral c, $z(c)$ is integral.

($\Rightarrow$) We will prove P is integral by proving that every rational supporting hyperplane contains an integer point (see Proposition 5.2). For any rational supporting hyperplane, $\pi^\top x = \pi_0$, we can scale π to an integral vector c and gcd(c)=1, and let $c = \lambda\pi$. By property of supporting hyperplane and hypothesis, we have $\lambda\pi_0 = z(c)$ is integral. Then we know there exists a integer point $\hat{x}$ that $c^\top \hat{x} = z(c)$, and we finish our proof. □

5.2.1 Totally Dual Integral System (TDI)

We first define the TDI, assume all data below (A, b, c) are rational, and consider two problem

$$(P) \quad \max\ c^\top x \qquad\qquad (D) \quad \min\ y^\top b$$
$$s.t.\ Ax \leq b \qquad\qquad s.t.\ y^\top A = c$$

Definition 5.3. *If for any $c \in \mathbb{Z}^n$, such that the LP (P) is finite, there exists an integer optimal solution to the dual LP (D), then $Ax \leq b$ is said to be TDI.*

Corollary 5.1. *If $Ax \leq b$ is TDI and b is integral, then $\{x \mid Ax \leq b\}$ is integral.*

Proof. Let c be an integral vector. Since $Ax \leq b$ is TDI, there exists an integral optimal solution $\hat{y}$. Then for integral b, the objective value of (D), e.g. v_D is integral. By strong duality, $c^\top x^* = \hat{y}^\top b$ is integral, then by Proposition 5.3, we claim that P is integral. □

Observe that if A is TU, then the system $Ax \leq b$ is TDI. Thus the notion of TDI generalizes the notion of TU systems. See Section 6.3.2 for an example for a integral polytope that is a TDI system and b being integral.

5.3 Suggested exercises

1. Provide a counterexample that demonstrates that the total unimodularity of the matrices A and B is not sufficient to guarantee that the composed matrix $[A\ B]$ is totally unimodular.

2. Let $S = \{S_1, \ldots, S_m\}$ be a family of subsets of a nonempty finite set $V := \{1, \ldots, n\}$, and let A denote the $|S| \times |V|$ incidence matrix of S, i.e.
$$A_{ij} = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{otherwise} \end{cases}$$
The family S is laminar if, for all $U, V \in S$ such that $U \cap V \neq \emptyset$, either $U \subseteq V$ or $V \subseteq I$. Show that A is Totally unimodular matrix.

3. Consider the problem:
$$\begin{aligned} \max \quad & \sum_{1 \le i < j \le n} c_{ij} x_i x_j - \sum_{i=1}^{n} d_i x_i \\ \text{s.t.} \quad & x \in \{0,1\}^n \end{aligned}$$
Assuming c is non-negative, show that the above problem can be solved in polynomial-time.

4. Let $G = (V, E)$ be a simple graph. Given a subset $V' \subseteq V$, $G[V'] = (V', E')$ denotes the graph where $E' \subseteq E$ is the subset of edges that have both end points in V'. A subset $\tilde{V} \subseteq V$ is called *perfectly matchable* if the graph $G[V']$ contains a perfect matching. The perfectly matchable subgraph polytope of G, denoted by P^{pm}, is the convex hull of of the incidence vectors of all subsets of V that are perfectly matchable.

   Suppose that G is a bipartite graph. Prove that
$$P^{pm} = \Big\{ x \in \mathbb{R}^{|V|} \,|\, \exists z \in \mathbb{R}^{|E|} \text{ s.t. } \sum_{e \in \delta(v)} z_e = x_v \ \forall v \in V, \\ 0 \le x_v \le 1 \ \forall v \in V,\ z_e \ge 0 \ \forall e \in E \Big\}.$$

5.
   - Let $A \in \mathbb{Z}^{m \times n}$. Suppose that for all $b \in \mathbb{Z}^m$ and $\hat{x} \in \mathbb{Z}^n_+$ and for all integers $k \ge 1$, with $A\hat{x} \le kb$, there are integral vectors $x^1, \ldots, x^k$ in $\{x \ge 0 \,|\, Ax \le b\}$ such that $\hat{x} = x^i + \cdots + x^k$. Prove that A is totally unimodular.

- Now prove the converse: Let A be a totally unimodular matrix. Let $P := \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$ and $b \in \mathbb{Z}^m$. Let $\frac{1}{k} \cdot \hat{x} \in P$ where $k \in \mathbb{Z}_+$ such that $\hat{x} \in \mathbb{Z}^n$. Prove that $\hat{x} = x^1 + x^2 + \cdots + x^k$ where $x^i \in P \cap \mathbb{Z}^n$ for all $i \in [k]$. [Hint: Prove by induction on k. For $k \geq 2$, examine the polytope: $\{x \mid 0 \leq x \leq \hat{x}, A\hat{x} - (k-1)b \leq Ax \leq b\}$]

6. Show that TU property is preserved under the following operations:
   (a) permuting rows and columns
   (b) taking the transpose
   (c) multiplying a row or column by -1
   (d) pivoting, i.e., replacing $\begin{bmatrix} a & c^\top \\ b & D \end{bmatrix}$ by $\begin{bmatrix} -a & ac^\top \\ ab & D - abc^\top \end{bmatrix}$ (a is a scalar)

7. Assuming $[A\ a]$ and $\begin{bmatrix} b^\top \\ B \end{bmatrix}$ are TU matrices. Then show that $\begin{bmatrix} A & ab^\top \\ 0 & B \end{bmatrix}$ is a TU matrix.

8. (a) Given scalars $b > 0$ and $a_j > 0$ for $j = 1, \ldots, n$, consider the $0,1$ knapsack set $K := \{x \in \{0,1\}^n \mid \sum_{i=1}^n a_i x_i \leq b\}$. A minimal cover is a set $C \subseteq \{1, \ldots, n\}$ such that:
   - $\sum_{j \in C} a_j > b$
   - $\sum_{j \in C \setminus \{i\}} a_j \leq b$ for all $i \in C$

   Consider the set
   $$K(C) := \{x \in \{0,1\}^n \mid \sum_{i \in C} \leq |C| - 1 \text{ for every minimal cover } C \text{for } K\}.$$
   Prove that $K = K(C)$

   (b) Consider the feasible region of a $0-1$ knapsack of the following form:
   $$S := \left\{ x \in \{0,1\}^n \mid \sum_{i=1}^n 2^{i-1} x_i \leq b \right\},$$
   where $2^{n-1} \leq b < 2^n$. Let the binary expansion of b be
   $$b = 2^{i_1 - 1} + 2^{i_2 - 1} + \cdots + 2^{i_r - 1} + 2^{n-1}.$$
   Let $I := \{i_1, i_2, \ldots, i_r, n\}$. For any $j \in \{1, \ldots, n\} \setminus I$ define the set $I_j := \{j\} \cup \{i \in I \mid i > j\}$.

(a) Prove that I_j defines a minimal cover for all $j \in \{1, \ldots, n\} \setminus I$.

(b) Prove that the cover inequalities from the different I_js put together forms an integral polytope, i.e., show that the following polytope is integral:

$$S(C) := \left\{ x \in [0,1]^n \mid \sum_{t \in I_j} x_t \leq |I_j| - 1 \ \forall \ j \in \{1, \ldots, n\} \setminus I \right\}.$$

9. Construct an example of a TDI system $Ax \leq b$, where A is not TU.

10. Show that for any rational linear system $Ax \leq b$ there is a positive rational number q such that $qAx \leq qb$ is TDI.

Chapter 6

Some Well-known Polytopes

6.1 Lot-sizing-problem

Recall the Lot-Sizing Problem introduced in the beginning of this class. At each period a demand needs to be met (the demand is known in advance). Periodically, the manufacturer can decide to produce or not to produce items (the production involves fixed cost). A holding cost is also incurred over the periods. The problem is to decide how many items to manufacture in each time period to minimize the sum of the setup cost and inventory cost. Consider the feasible set of the uncapacitated Lot-Sizing problem:

$$
\begin{aligned}
X_n = \{(x, y, s) \in \mathbb{R}^{3n} : \quad & s_{t-1} + x_t = d_t + s_t, && t = 1, ..., n, \\
& x_t \leq d_{1n} y_t, && t = 1, ..., n, \\
& x_t, s_t \geq 0, y_t \in \{0, 1\}, && t = 1, ..., n\}
\end{aligned}
$$

Where $\{1, 2, ..., n\}$ is the time index, d_t is the demand in period t (We use notation $d_{ij} := \sum_{t=i}^{j} d_t$), s_t is the amount in stock at the end of period t (We assume $s_0 = s_n = 0$ in this lecture), x_t is the amount produced in period t, $y_t \in \{0, 1\}$ indicates whether a set-up cost must be incur in period t, which must have the value 1 if $x_t > 0$.
Notice that it is possible to eliminate the "stock" variables s_t from the

description of X_n, and giving $X \subseteq \mathbb{R}^{2n}$ defined as follows:

$$\begin{aligned}
X = \{(x,y) \in \mathbb{R}^{2n} : \quad & \sum_{u=1}^{t} x_u \geq d_{1t}, && t = 1, ..., n, && (6.1)\\
& \sum_{u=1}^{n} x_u = d_{1n}, && && (6.2)\\
& x_t \leq d_{1n} y_t, && t = 1, ..., n, && (6.3)\\
& x_t \geq 0, && t = 1, ..., n, && (6.4)\\
& y_t \in [0,1], && t = 1, ..., n, && (6.5)\\
& y_t \in \mathbb{Z}, && t = 1, ..., n, \} && (6.6)
\end{aligned}$$

In other words, at each time period, the production by so far must be greater than or equal to the demand by so far (1). The total production and total demand over the time period must be equal (2). Positive production incurs fixed costs and must be less than or equal to the total demand (3). demand must be non-negative (4) and fixed cost variables are zero-one integers (5)(6).

6.1.1 Lot-Sizing inequalities

Definition 6.1. *Let $\ell \in \{1, 2, ..., n\}$, $L := \{1, 2, \ldots, \ell\}$, and $S \subseteq L$. The **Lot-Sizing inequality** is given by*

$$\sum_{i \in S} x_i + \sum_{i \in L \setminus S} d_{i\ell} y_i \geq d_{1\ell} \tag{6.7}$$

Theorem 6.1. *For any $1 \leq \ell \leq n$, $L = \{1, 2, \ldots, \ell\}$, and $S \subseteq L$, the Lot-Sizing inequalities are valid for X.*

Proof. Let $(\hat{x}, \hat{y})$ be an arbitrary point in X.
Case 1: If $\hat{y}_i = 0$, $\forall i \in L \setminus S$, then

$$\sum_{i \in S} \hat{x}_i + \sum_{i \in L \setminus S} d_{i\ell} \hat{y}_i = \sum_{i \in S} \hat{x}_i = \sum_{i \in L} \hat{x}_i \geq d_{1\ell}.$$

Case 2: If on the contrary $\exists i \in L \setminus S$ such that $\hat{y}_i = 1$, let $k = \arg\min_i \{i \in$

$L \setminus S : \hat{y}_i = 1\}$, then

$$\begin{aligned}
\sum_{i \in S} \hat{x}_i + \sum_{i \in L \setminus S} d_{i\ell} \hat{y}_i &\geq \sum_{i=1}^{k-1} \hat{x}_i + \sum_{i \in L \setminus S} d_{i\ell} \hat{y}_i && \text{(a)} \\
&\geq \sum_{i=1}^{k-1} \hat{x}_i + d_{k\ell} && \text{(b)} \\
&\geq d_{1,k-1} + d_{k\ell} \\
&= d_{1\ell}
\end{aligned}$$

(a) follows from

$$\begin{aligned}
\sum_{i=1}^{k-1} x_i &= \underbrace{\sum_{i \in \{1,2,\ldots,k-1\} \cap (L \setminus S)} \hat{x}_i}_{= 0 \text{ by definition of } k} + \sum_{i \in \{1,2,\ldots,k-1\} \cap S} \hat{x}_i \\
&\leq \sum_{i \in S} \hat{x}_i
\end{aligned}$$

(b) follows from

$$\sum_{i \in L \setminus S} d_{i\ell} \hat{y}_i \geq d_{k\ell} \hat{y}_k = d_{k\ell}.$$

□

Theorem 6.2. *If $d_t \geq 0, \forall t \in \{1, 2, ..., n\}$, the Lot-Sizing inequalities are facet defining for conv(X) whenever $\ell < n, 1 \in S$ and $L \setminus S \neq \emptyset$. All facets are distinct.*

Proof. Since the dimension of $conv(X)$ is $(2n-2)$ (we will prove this in the next section), the dimension of a facet of $conv(X)$ is $(2n-3)$. The idea is to find $(2n-2)$ affinely independent feasible points that are tight at the Lot-Sizing inequality. The detailed proof can be found in [3]. □

6.1.2 The Separation Problem for Lot-Sizing Inequalities

Given (x^*, y^*) satisfying (1)-(5), either find a Lot-Sizing inequality cutting off the point, or show that all the inequalities are satisfied.

Theorem 6.3. *The following algorithm solves the separation problem for Lot-Sizing inequalities.*

Algorithm 1 Lot-Sizing Separation

for $\ell = 1$ **to** n **do**
  $\alpha_\ell := \sum_{j=1}^{\ell} \min(x_j^*, d_{j\ell} y_j^*)$.
  if $\alpha_\ell < d_{1\ell}$ **then**
    Output $L = \{1, 2, ..., \ell\}$ and $S = \{j \in L : x_j^* \leq d_{j\ell} y_j^*\}$
  end if
end for

Proof. First we show that if the algorithm produces an output, the Lot-Sizing inequality corresponding to the output cuts off the point. By definition of S, we have

$$\sum_{j \in S} x_j^* + \sum_{j \in L \setminus S} d_{j\ell} y_j^* = \sum_{j=1}^{\ell} \min(x_j^*, d_{j\ell} y_j^*) = \alpha_\ell < d_{1\ell}$$

Conversely let us show that if there exists a Lot-Sizing inequality (with parameters ℓ, L and S) cutting off the point, the pair (L, S) will be an output of the algorithm. By definition of α_ℓ, we have

$$\alpha_\ell = \sum_{j=1}^{\ell} \min(x_j^*, d_{j\ell} y_j^*) \leq \sum_{j \in S} x_j^* + \sum_{j \in L \setminus S} d_{j\ell} y_j^* < d_{1\ell}$$

□

6.1.3 Linear inequality description of the convex hull

Let

$$\mathscr{P} = \{(x, y) \in \mathbb{R}^{2n} : (x, y) \text{ satisfies (1)-(5) and (7)}, \forall 1 \leq \ell \leq n, L = \{1, 2, ..., \ell\}, \text{ and } S \subseteq L\}$$

We are going to prove that $\mathscr{P} = conv(X)$ under the condition of $d_t > 0, \forall t \in \{1, 2, ..., n\}$. To show this, we need the following proposition, which you will prove in your homework.

Proposition 6.1. *Suppose $A, B \in \mathbb{R}^n$ are two polytopes with the following properties:*
(i) $A \subseteq B$

(ii) dim(A) = dim(B)
(iii) Let M(c) be the set of optimal solutions when optimizing $c^T x$ *over A. For all* $c \in \mathbb{R}^n$ *such that the objective value is finite and* $M(c) \neq A$, *M(c) is completely contained in exactly one proper face of B.*
Then

$$A = B.$$

Theorem 6.4. *When* $d_t > 0, \forall t \in \{1, 2, ..., n\}, \mathscr{P} = conv(X)$.

It is clear that both $\mathscr{P}$ and $conv(X)$ are polytopes. To apply Proposition 1 we need to prove that conditions (i),(ii) and (iii) hold for A being $conv(X)$ and B being $\mathscr{P}$. The proof is relatively long, we break it into 3 pieces so that it looks more organized.

Lemma 6.1. $conv(X) \subseteq \mathscr{P}$.

Proof. Since the Lot-Sizing inequalities are valid for X, they are also valid for $conv(X)$. We have $conv(X) \subseteq \mathscr{P}$. □

Lemma 6.2. $dim(conv(X)) = dim(\mathscr{P}) = 2n - 2$.

Proof. The number of variables in $\mathscr{P}$ is $2n$. As $d_1 > 0$, all feasible solutions of $\mathscr{P}$ satisfy:

$$\sum_{i=1}^{n} x_i = d_{1n} \text{ and } y_1 = 1$$

The first equality follows from constraints (2), the second equality is obtained by observing that the Lot-Sizing inequality $d_1 y_1 \geq d_1$ ($\ell = 1$ and $S = \emptyset$ in this case) along with the constraint $y_1 \leq 1$ implies $y_1 = 1$. We have $dim(\mathscr{P}) \leq 2n - 2$.
Now we show that $dim(conv(X)) \geq 2n - 2$, it is enough to exhibit $2n - 1$ *affinely independent* points in X (i.e. no non-trivial combination of points with zero sum of coefficients is zero). The first n points are of the form:

$$\begin{bmatrix} x & y \\ d_{1n} & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} x & y \\ d_1 & 1 \\ d_{2n} & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \ldots, \quad \begin{bmatrix} x & y \\ d_1 & 1 \\ d_2 & 1 \\ d_3 & 1 \\ \vdots & \vdots \\ d_{n-1} & 1 \\ d_n & 1 \end{bmatrix}$$

The last $n-1$ points are of the form:

$$\begin{bmatrix} x & y \\ d_{1n} & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} x & y \\ d_{1n} & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \ldots, \quad \begin{bmatrix} x & y \\ d_{1n} & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

We conclude that

$$\begin{aligned} & 2n-2 \leq dim(conv(X)) \leq dim(\mathscr{P}) \leq 2n-2 \\ \implies & dim(conv(X)) = dim(\mathscr{P}) = 2n-2 \end{aligned}$$

. □

Lemma 6.3. *Let $M(p,q)$ ($p,q \in \mathbb{R}^n$ arbitrary) be the set of all optimal solutions to the problem minimizing $\sum_{t=1}^n p_t x_t + \sum_{t=1}^n q_t y_t$ such that $(x,y) \in conv(X)$ with $M(p,q) \neq conv(X)$, then $M(p,q)$ is contained in exactly one proper face of $conv(X)$.*

Proof. First we observe that as $\sum_{t=1}^n x_t = d_{1n}$, we can add any multiple of this constraint to the objective function without modifying the set $M(p,q)$. Thus by adding to the objective $(-\min_t p_t)(\sum_{u=1}^n x_u)$, we can assume without loss of generality that $\min_t p_t = 0$. In addition, since we have $y_1 = 1$, we can assume that $q_1 = 0$. Notice that now p_t are all non-negative, but q_t can be negative for $t \geq 2$.
Suppose $q_t < 0$ for some $t \geq 2$. For any feasible solution with $y_t = 0$, if we change y_t from 0 to 1 while keeping the value of other variables, the new solution is still feasible. But the new feasible solution would make the objective strictly smaller. We must have

$$M(p,q) \subseteq \{(x,y) : y_t = 1\}. \tag{6.8}$$

We suppose from now on that $q_t \geq 0$ for $t \geq 2$. Let $\ell = \arg\max_t \{p_t + q_t > 0\}$, as $(p,q) \neq (0,0)$ (otherwise $M(p,q) = conv(X)$), we must have $1 \leq \ell \leq n$. Suppose that $p_k = q_k = 0$ for some $k < \ell$, which means we can produce as much as we can in the k^{th} time period with zero cost so that we do not need to produce anything in the rest time periods. Since $p_\ell + q_\ell > 0$, we must have

$$M(p,q) \subseteq \{(x,y) : x_\ell = 0\}. \tag{6.9}$$

Otherwise we have $p_t + q_t > 0, \forall t \leq \ell$ and $p_t + q_t = 0, \forall t > \ell$. Now let $L = \{1, 2, ..., \ell\}$ and $S = \big\{t \in L : p_t > 0\big\}$, notice that $1 \in S$ since we have assumed that $q_1 = 0$. We show that

$$M(p,q) \subseteq \{(x,y) : \sum_{t\in S} x_t + \sum_{t\in L\setminus S} d_{t\ell} y_t = d_{1\ell}\}. \tag{6.10}$$

Consider an optimal solution (x^*, y^*). Let $\tau := \min\{t \in L \setminus S : y_t^* = 1\}$. This is under the assumption that $L\setminus S \neq \emptyset$ and $\exists t \in L\setminus S$ such that $y_t^* \neq 0$, we will deal with these special cases later.
As $p_\tau = 0$ by definition of S, one can produce an amount $d_{\tau l}$ or more at zero variable cost in time period τ. Also as $p_t + q_t > 0$ for all $t \in L$ with $t > \tau$, in an optimal solution we must have $x^* = y^* = 0$ for all $t \in L$ with $t > \tau$. This is true because otherwise a strictly optimal solution would be obtained by reducing x_t^* and/or y_t^* and increasing x_τ^*, which is a contradiction to the optimality of (x^*, y^*). So we have

$$\sum_{t\in L\setminus S} d_{t\ell} y_t^* = \underbrace{\sum_{t\in L\setminus S: t<\tau} d_{t\ell} y_t^*}_{=0 \text{ by definition of } \tau} + d_{\tau\ell} y_\tau^* + \underbrace{\sum_{t\in L\setminus S: t>\tau} d_{t\ell} y_t^*}_{=0 \text{ since } x_t^* = y_t^* = 0,\, \forall t \in L, t>\tau} = d_{\tau\ell} y_\tau^* = d_{\tau\ell}$$

And $\sum_{t\in S: t\geq\tau} x_t^* = 0$ (notice that $\tau \notin S$). As $x_t^* = y_t^* = 0$ for all $t \in L \setminus S$ with $t < \tau$, $\sum_{t=1}^{\tau-1} x_t^* = \sum_{t\in S:t<\tau} x_t^* \geq d_{1,\tau-1}$. But as $p_t > 0$ for all periods $t \in S$ with $t < \tau$, a solution can only be optimal if $\sum_{t\in S:t<\tau} x_t^* = d_{1,\tau-1}$. This is true because otherwise a strictly better solution can be obtained by reducing $\sum_{t\in S:t<\tau} x_t^*$ and increasing x_τ^*, which leads to a contradiction. We have shown that

$$\sum_{t\in S:t<\tau} x_t^* + \sum_{t\in S:t\geq\tau} x_t^* + \sum_{t\in L\setminus S} d_{t\ell} y_t^* = d_{1,\tau-1} + 0 + d_{\tau\ell} = d_{1\ell}. \tag{6.11}$$

Since the above equality holds for any $(x^*, y^*) \in M(p,q)$, we obtain (10).
In the special case where $L\setminus S = \emptyset$ (that is $L = S$), notice that since $\min_t p_t = 0$, we must have $\ell < n$. Then $\sum_{t\in S} x_t^* + \sum_{t\in L\setminus S} d_{t\ell} y_t^* = \sum_{t=1}^{\ell} x_t^* = d_{1\ell}$, because production in period $\ell + 1$ has zero cost.

In the special case where $y_t^* = 0$, $\forall t \in L \setminus S$, by the same argument we have $\sum_{t \in S} x_t^* + \sum_{t \in L \setminus S} d_{t\ell} y_t^* = \sum_{t=1}^{\ell} x_t^* = d_{1\ell}$.
Now it can be readily checked that all the faces used above in the proof ((8),(9) and (10)) are proper faces of $\mathscr{P}$. $\square$

6.2 Perfect Matching

Recall that a matching in an undirected graph $G = (V, E)$ is a set $M \subseteq E$ of pairwise disjoint edges, where an edge is viewed here as a set of two distinct nodes. In other words, a matching is an independent edge set. A matching is *perfect* if it covers every node of the graph (that is, every node is contained in exactly one edge of the matching). A basic problem in combinatorial optimization is the maximum cardinality matching problem, that is, finding a matching of G of maximum cardinality. Obviously a *perfect matching* of G has maximum cardinality, but in general a perfect matching might not exist. Specifically, perfect matchings are only possible on graphs with an even number of vertices.

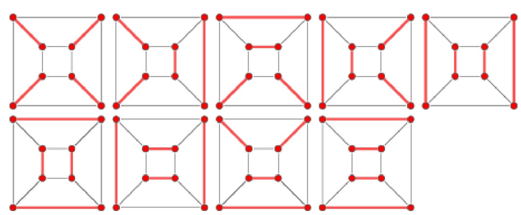


Figure 6.1: The nine perfect matchings of the cubical graph.

Recall that the *perfect matching polytope* of G is the convex hull of all characteristic vectors of perfect matchings. In this lecture, we will provide a representation for the perfect matching polytope. This representation is provided as a theorem, namely, the *Perfect Matching Polytope Theorem* or *Edmonds' Matching Polyhedron Theorem*. We will devote this lecture to proving this theorem.

Suppose that $G = (V, E)$ is an undirected graph with $|V|$ even (we will allow multiple edges between any two vertices). Let $S(G)$ be the perfect matching polytope, i.e., $S(G)$ = convex hull of perfect matchings of the graph G (in which multiple edges between nodes are allowed). For any $V' \subseteq V$, we define $\delta(V')$ as the set of edges of G that intersect V' in exactly one point.

Definition 6.2.

$$P(G) := \{x \in [0,1]^E \mid x_e := x(e) \text{ satisfies the conditions (6.12a)-(6.12c)}\}$$

$$x_e \geq 0,\ \forall e \in E, \tag{6.12a}$$

$$\sum_{e \in \delta(\{v\})} = 1,\ \forall v \in V, \tag{6.12b}$$

$$\sum_{e \in \delta(U)} \geq 1,\ \forall U \subset V,\ \text{where } |U| \text{ is odd, and } 3 \leq |U| \leq |V| - 3. \tag{6.12c}$$

Theorem 6.5 (Edmonds' Matching Polyhedron Theorem)**.** *The perfect matching polytope of a graph $G = (V, E)$ with $|V|$ even is determined by the inequalities (6.12a)-(6.12c), i.e. $S(G) = P(G)$.*

Proof. $S(G) \subseteq P(G)$ is obvious, since $P(G)$ is convex, and every perfect matching satisfies the conditions (6.12a)-(6.12c).

We then prove $P(G) \subseteq S(G)$ by contradiction. Suppose that there exists some G such that $P(G) \not\subseteq S(G)$. Let $\hat{G} = (\hat{V}, \hat{E})$ be a smallest graph (in terms of $|\hat{V}| + |\hat{E}|$) such that $P(\hat{G}) \not\subseteq S(\hat{G})$ Therefore, at least one extreme point, say $\hat{x}$, of $P(\hat{G})$ is not in $S(\hat{G})$.

Claim 1 : $1 > \hat{x}_e > 0,\ \forall e \in \hat{E}$.

Proof. Suppose $\hat{x}_e = 1$ for some $e \in \hat{E}$. Then by deleting this edge e and its incident vertices from $\hat{G}$, we can obtain a smaller counterexample. Similarly, suppose $\hat{x}_e = 0$, then by deleting this edge e, we can obtain a smaller counterexample. □

Claim 2 : $deg(v) \geq 2,\ \forall v \in \hat{V}$.

Proof. For any $v \in \hat{V}$, there exist $e, e' \in \delta(\{v\})$ with $e \neq e'$ such that $\hat{x}_e, \hat{x}_{e'} > 0$ (due to Claim 1 and constraint (6.12b)). □

Claim 3 : $|\hat{E}| \geq |\hat{V}|$

Proof. This is because $\sum_{v \in \hat{V}} \deg(v) = 2|\hat{E}|$, and $2|\hat{V}| \leq \sum_{v \in \hat{V}} \deg(v)$ by claim 2. □

Claim 4 : Suppose that $|\hat{E}| = |\hat{V}|$; then Theorem 6.5 is correct.

Proof. Suppose $|\hat{E}| = |\hat{V}|$, it implies $\deg(v) = 2$ for all $v \in \hat{V}$; in other words, $\hat{G}$ is a union of cycles. There are two cases:

Case 1: All the cycles are of even cardinality. Then Theorem 6.5 holds trivially, as constraints (6.12a) and (6.12b) already determine the convex hull $S(\hat{G})$, i.e., $S(\hat{G}) = P(\hat{G})$. For example, given $|\hat{V}| = 4$, then $P(\hat{G}) = \{(a, 1-a, a, 1-a) \in [0,1]^4 : a \in [0,1]\}$. The only extreme points of $P(\hat{G})$ are $(0,1,0,1)$ and $(1,0,1,0)$ that lie in $S(\hat{G})$. This is because for $0 < a < 1$, we have

$$(a, 1-a, a, 1-a) = \frac{1}{2}(a-\epsilon, 1-a+\epsilon, a-\epsilon, 1-a+\epsilon) + \frac{1}{2}(a+\epsilon, 1-a-\epsilon, a+\epsilon, 1-a-\epsilon),$$

where $\epsilon = \min\{a, 1-a\} > 0$. This means $P(\hat{G}) \subseteq S(\hat{G})$. We can easily generalize the above argument to the cyclic graph $\hat{G}$ with even $|\hat{V}|$. [Alternatively this follows from the fact that the matrix corresponding to the LP relaxation is TU.]

Case 2: There exists an odd cycle in $\hat{G}$. Since the odd number of vertices implies no perfect matching, then $S(\hat{G})$ and $P(\hat{G})$ are both empty.

□

Claim 5 : We consider the case $|\hat{E}| > |\hat{V}|$. Note that $\hat{x}$ is $|\hat{E}|$-dimensional and it is an extreme point of $P(\hat{G})$. However, at most $|\hat{V}|$ inequalities are binding at $\hat{x}$ among the constraints (6.12a) and (6.12b) (due to Claim 1). Then there exists some $\tilde{U}$ corresponding to inequality (6.12c) (with $|\hat{U}|$ odd and $3 \leq |\hat{U}| \leq |\hat{V}| - 3$), which is binding at $\hat{x}$ such that

$$\sum_{e \in \delta(\tilde{U})} \hat{x}_e = 1. \tag{6.13}$$

Definition 6.3 (Vertex Contraction)**.** *The contraction of a set of vertices (or nodes) $V' = \{v_1, ..., v_k\}$ in a graph $G = (V, E)$ $(V' \subseteq V)$ produces a graph in which the nodes in V' are replaced with a single node v^* such that v^* is adjacent to the union of the nodes to which V' were originally adjacent. In vertex contraction, it doesn't matter if $v_i, v_j \in V'$ are connected by an edge; if they are, the edge is simply removed upon contraction. See* `http://mathworld.wolfram.com/VertexContraction.html` *for more details.*

Next we construct two graphs by contracting $\tilde{U}$ and $V\backslash\tilde{U}$ respectively. Suppose that $\tilde{U}$ contracts to v^{*1}, and $V\backslash\tilde{U}$ contracts to v^{*2}. We define

$$\hat{G}^1 := (\hat{V}^1, \hat{E}^1),$$

where $\hat{V}^1 = \tilde{U} \cup \{v^{*2}\}$, $\hat{E}^1 = \{(v', v'') : v', v'' \in \tilde{U}, (v', v'') \in \hat{E}\} \cup \{(v', v^{*2}) : v' \in \tilde{U}, v'' \in V\backslash\tilde{U}, (v', v'') \in \hat{E}\}$. Similarly, we define

$$\hat{G}^2 := (\hat{V}^2, \hat{E}^2),$$

where $\hat{V}^2 = \{v^{*1}\} \cup V\backslash\tilde{U}$, $\hat{E}^2 = \{(v', v'') : v', v'' \in V\backslash\tilde{U}, (v', v'') \in \hat{E}\} \cup \{(v^{*1}, v'') : v' \in \tilde{U}, v'' \in V\backslash\tilde{U}, (v', v'') \in \hat{E}\}$. Note that both $\hat{G}^1$ and $\hat{G}^2$ may contain multiple edges between two nodes.

Define $\hat{x}_1$ and $\hat{x}_2$ as the projections of $\hat{x}$ onto (the edges of) $\hat{G}^1$ and $\hat{G}^2$, respectively. Denote by $\delta_{\hat{G}^i}(V')$ the set of edges of $\hat{G}^i$ that intersect $V' \subseteq \hat{V}^i$ in exactly one point for $i = 1, 2$.

Claim 6 : $\hat{x}^1 \in P(\hat{G}^1)$ and $\hat{x}^2 \in P(\hat{G}^2)$.

Proof. We show $\hat{x}^1 \in P(\hat{G}^1)$ by checking the constraints (6.12a)-(6.12c) corresponding to $P(\hat{G}^1)$. We can also show $\hat{x}^2 \in P(\hat{G}^2)$ in a parallel way.

Constraint (1): By the definition of $\hat{x}$, $\hat{x}^1_{e^1} = \hat{x}_e > 0$ for any $e \in \hat{E}^1$, where e^1 is the projection of e on $\hat{G}^1$.

Constraint (2): It is easy to see $\sum_{e^1 \in \delta_{\hat{G}^1}(\{u\})} \hat{x}^1_{e^1} = \sum_{e \in \delta(\{u\})} \hat{x}_e = 1$ if $u \in \tilde{U}$. On the other hand, $\sum_{e^1 \in \delta_{\hat{G}^1}(\{v^{*2}\})} \hat{x}_{e^1} = 1$ due to (6.13). Note that

$$\sum_{e^1 \in \delta_{\hat{G}^1}(\{v^{*2}\})} \hat{x}_{e^1} = \sum_{e^1 \in \delta_{\hat{G}^2}(\{v^{*1}\})} \hat{x}_{e^2} = \sum_{e \in \delta(\tilde{U})} \hat{x}_e.$$

where e^i is the projection of e on $\hat{G}^i$ for $i = 1, 2$.

Constraint (3): We first pick any $U \subseteq \tilde{U}$ satisfying that $|U|$ is odd and $3 \leq |U| \leq |\hat{V}^1| - 3$: it is easy to see that $\sum_{e^1 \in \delta_{\hat{G}^1}(U)} \hat{x}^1_{e^1} = \sum_{e \in \delta(U)} \hat{x}_e \geq 1$. Then we consider the case $U = \{u^1, ..., u^k, v^{*1}\}$ with k even. Note that we require $2 \leq k \leq |\hat{V}| - 6$; otherwise, $|U| = k + 1 \notin [3, |\hat{V}^1| - 3]$ (recall that $|\hat{V}^1| \leq |\hat{V}| - 2$); we will revisit this point in Claim 7). Then

$$\sum_{e^1 \in \delta_{\hat{G}^1}(U)} \hat{x}^1_{e^1} = \sum_{e \in \delta(\bar{U})} \hat{x}_e \geq 1,$$

where $\bar{U} = \{u^1, ..., u^k\} \cup (V \backslash \tilde{U})$. Then the inequality holds since $3 \leq |\bar{U}| \leq (|\hat{V}| - 6) + 3 = |\hat{V}| - 3$. $\square$

Claim 7 : $\hat{x}^1 \in S(\hat{G}^1), \hat{x}^2 \in S(\hat{G}^2)$

Proof. Otherwise, $\hat{G}^1$ and $\hat{G}^2$ is a smaller counterexample, which contradicts our "smallest" assumption on $\hat{G}$ with the property $P(\hat{G}) \not\subseteq S(\hat{G})$. To see this, note that $\hat{G}^1$ is a smaller graph c.f. $\hat{G}$, since the number of vertices is reduced by at least 2, and the number of edges is also reduced; we can apply similar argument on $\hat{G}^2$. $\square$

Claim 7 means that $\hat{x}^1$ and $\hat{x}^2$ can be represented as convex combinations of perfect matchings in $\hat{G}^1$ and $\hat{G}^2$, respectively, i.e.,

$$\hat{x}^1 = \sum_{i=1}^{t_1} \lambda_i x^{m'_i},\ \hat{x}^2 = \sum_{j=1}^{t_2} \mu_j x^{m''_j},$$

where $\lambda_i, \mu_j \geq 0, \sum_{i=1}^{t_1} \lambda_i = 1, \sum_{j=1}^{t_2} \mu_j = 1, x^{m'_i}$ are $0-1$ vectors corresponding to some perfect matching m'_i in $\hat{G}^1$ for $i = 1, ..., t_1$, and $x^{m''_j}$ are $0-1$ vectors corresponding to some perfect matching m''_j in $\hat{G}^2$ for $j = 1, ..., t_2$.

We will show that these decompositions can be glued together to represent $\hat{x}$ as a convex combination of perfect mappings in $\hat{G}$, contradicting our assumption that $\hat{x} \notin S(\hat{G})$, i.e., $P(\hat{G}) \not\subseteq S(\hat{G})$.

Claim 8 : $\lambda' s$ and $\mu' s$ are rational numbers.

Proof. Note that $\hat{x}$ is rational because it is an extreme point of a rational polytope $P(\hat{G})$. Then $\hat{x}^1$ and $\hat{x}^2$ are both rational because they are projections of $\hat{x}$ on $\hat{G}^1$ and $\hat{G}^2$. Similarly, $\{x^{m'_i}\}$ and $\{x^{m''_j}\}$ are rational, since they are extreme points of rational polytopes $P(\hat{G}^1)$ and $P(\hat{G}^2)$, respectively. Therefore, $\lambda' s$ and $\mu' s$ should be rational numbers. $\square$

Claim 9 : We may assume that

$$\hat{x}^1 = \frac{1}{K} \sum_{i=1}^{K} x^{m'_i},\ \hat{x}^2 = \frac{1}{K} \sum_{j=1}^{K} x^{m''_j},$$

where K is a natural number (e.g., the least common multiple of the denominators of $\lambda's$ and $\mu's$ by representing them as fractions). Note that $\{m'_i\}_{i=1}^K$ (resp., $\{m''_j\}_{j=1}^K$) may contain repetitions of the extreme points of $S(\hat{G}^1)$ (resp. $S(\hat{G}^2)$).

Pick any $e = (v', v'') \in \delta(\tilde{U})$, i.e., $(v', v'') \in \hat{E}$ with $v' \in \tilde{U}$ and $v'' \in V \backslash \tilde{U}$. Then by construction of $\hat{G}^1$ and $\hat{G}^2$ we have $\hat{x}^1_{e^1} = \hat{x}^2_{e^2} = \hat{x}_e$, where $e^1 = (v', v^{*2})$ and $e^2 = (v^{*1}, v'')$. We can further assume that $x_e^{m'_i} = x_e^{m''_j}$ if $i = j$. This is because with a fixed m'_i there is exactly one $e^1 = (v', v^{*2}) \in \delta_{\hat{G}^1}(\tilde{U})$ such that $x_e^{m'_i} = 1_{\{e=e^1\}}$; with a fixed m''_j there is exactly one $e^2 = (v^{*1}, v'') \in \delta_{\hat{G}^2}(\hat{V} \backslash \tilde{U})$ such that $x_e^{m'_i} = 1_{\{e=e^2\}}$. Thus we can arrange the order of $\{m'_i\}_{i=1}^K$ such that $x_e^{m'_i} = x_e^{m''_i}$ for $e \in \delta(\tilde{U})$ and all $i = 1, ..., K$. It is straightforward to see that $m_i = m'_i \cup m''_i$ (after the rearrangement) is a perfect matching in $\hat{G}$.

Then we have the representation $\hat{x} = \frac{1}{K} \sum_{i=1}^K x^{m_i}$ (by checking $\hat{x}_e = \frac{1}{K} \sum_{i=1}^K x_e^{m_i}$ for all $e \in \hat{E}$), which is a contradiction to $\hat{x} \notin S(\hat{G})$ (or equivalently, $P(\hat{G}) \not\subseteq S(\hat{G})$).

□

6.3 Matroid and polymatroid

First, we define a matroid and introduce the notion of rank.

Definition 6.4 (Matroid)**.** *A matroid is a pair $\mathcal{M} = \{S, \mathcal{C}\}$ consisting of a finite set S and a nonempty collection $\mathcal{C}$ of subsets of S (i.e. $\mathcal{C} \subseteq 2^S$) such that*

1. *If $I \in \mathcal{C}$ and $J \subseteq I$ then $J \in \mathcal{C}$.*
2. *If $I, J \in \mathcal{C}$ and $|I| < |J|$ then there exists $x \in J \backslash I$ such that $I \cup \{z\} \in \mathcal{C}$.*

The elements of $\mathcal{C}$ are called *independent sets*, and elements in $2^S \setminus \mathcal{C}$ are called *dependent sets*. We refer to 1 as the independence property and 2 as the exchange property.

Definition 6.5 (Base)**.** *Given some $U \subseteq S$, a set $I \subseteq U$ is called a base of U if I is a maximal (by inclusion) independent set contained in U. The bases of a matroid are the bases of S.*

Note that a matroid is a generalization of the notion of linear independence for vector spaces. To see this, let S be the set of columns of a matrix and let $\mathcal{C}$ be the sets of linearly independent columns of the matrix. The independence property is clear since a subset of a linearly independent set of vectors is also linearly independent. The exchange property holds since, if I, J are both sets of linearly independent vectors such that $|I| < |J|$, the span of the vectors of I cannot be the same as the span of the vectors of J, else J is would not be a linearly independent set. That means there must be some $x \in J$ such that x is not in the span of the vectors of I, so $I \cup \{x\}$ is a linearly independent set.

As another example of a matroid, consider a graph $G = (V, E)$. Let the base set $S = E$ and let $\mathcal{C}$ be the subsets of edges which do not have cycles, that is, forests on the edges of G. Then $\mathcal{M} = \{S, \mathcal{C}\}$ is a matroid.

As with vector spaces, all bases have the same cardinality.

Proposition 6.2. *Let $\mathcal{M} = \{S, \mathcal{C}\}$ be a matroid and $U \subseteq S$. Then the cardinality of any base of U is the same.*

Proof. Aiming towards contradiction, suppose $B^1 \subseteq U$ and $B^2 \subseteq U$ are bases of U where $|B^1| < |B^2|$. By the exchange property, there exists $z \in B^2 \setminus B^1$ such that $B^1 \cup \{z\} \in \mathcal{C}$. But $B^1 \cup \{z\} \subseteq U$. This means B^1 was not maximal by inclusion and hence was not a base, a contradiction. □

Based on the above Proposition, we can now make the following definition.

Definition 6.6 (Rank)**.** *The cardinality of a base of U is called the rank of U, which we denote $r(U)$. The rank of a matroid is the rank of S.*

6.3.1 Optimization over Matroids

Next, we consider the maximum-weight matroid problem. Let $\mathcal{M} = \{S, \mathcal{C}\}$ be a matroid and let $w : S \to \mathbb{R}^+$ be a function which assigns a weight to every element of S. For ease of notation, we say $w(U) = \sum_{u \in U} w(u)$. Then we can formulate our optimization problem as

$$\begin{aligned} \max_{U \subseteq 2^S} \quad & w(U) \\ \text{s.t.} \quad & U \in \mathcal{C} \end{aligned} \tag{6.14}$$

Greedy Algorithm

We can solve problem (6.14) with a greedy algorithm. We start with an empty set. At each iteration, we add the maximum-weight element from the set of elements that we have not yet selected such that our set remains independent. Formally, we have:

- For $j = 1, \ldots, r(S)$
- Find $y \in \operatorname{argmax}\{w(u) : u \in S \setminus I \text{ and } I \cup \{u\} \in \mathcal{C}\}$
- $I \leftarrow I \cup \{y\}$
- End-for

Note that, in the case of finding a maximum-weight spanning forest, this is exactly Kruskal's algorithm.

Correctness of the Greedy Algorithm

Theorem 6.6. *The greedy algorithm is correct.*

Proof. First, we know that $r(S)$ is finite, so the algorithm will terminate. We must show that the algorithm results in the optimal solution. We will say that a set $I \in C$ is a "good set" if I is contained in an optimal base of S. We will show that at every step of the algorithm, our set I is a good set. This shows our claim since the algorithm will terminate with a good set of cardinality $r(S)$, so I will itself be an optimal base. We proceed by induction on $|I|$.

The base case where $|I| = 0$ is trivial since $I = \emptyset$ is clearly a subset of any optimal base of S.

Now, assume that I is good. We must show that $I \cup \{y\}$ is good, where y is obtained in one iteration of the for loop. Let B be an optimal base such that $I \subseteq B$. Two cases arise depending on whether or not $y \in B$:

1. If $y \in B$ then $I \cup \{y\} \in B$, so $I \cup \{y\}$ is also good, as required.

2. We now consider the case where $y \notin B$. We will show that there exists $z \in B$ such that $B' := (B \cup \{y\}) \setminus \{z\}$ is a base of S. By assumption, $I \subseteq B$, and by the algorithm, $I \cup \{y\} \in \mathcal{C}$. Thus, if $|I \cup \{y\}| = |B|$, let $B' = I \cup \{y\}$, and we are done.

   Now suppose that $|I \cup \{y\}| < |B|$. Then, by the exchange property, there exists $v \in B \setminus (I \cup \{y\})$ such that $I \cup \{y\} \cup \{v\} \in \mathcal{C}$. We can repeat this process $k := |B| - |I \cup \{y\}|$ times until we have $B' = I \cup \{y\} \cup \{z_1\} \cup \cdots \cup \{z_k\} \in \mathcal{C}$, where $z_i \in B$ for all $1 \leq i \leq k$. We know that B' is a base since it has the same cardinality as B, and since y is the only "outsider" in B', we know that there is exactly one element $v \in B$ such that $v \notin B'$. Thus, B' is the desired base.

Last, we show that $w(B') \geq w(B)$. To see this, note that

$$w(B') - w(B) = w(y) - w(v) \geq 0, \tag{6.15}$$

where the inequality holds by the greediness of the algorithm: Since $v \notin B'$, $v \notin I$, when y was chosen, v was also available, but not chosen. This implies that the weight of y was at least as much as that of v. By (6.15), we see that B' is also an optimal base, so $I \cup \{y\}$ is a good set.

Thus, in both cases we found that $I \cup \{y\}$ is a good set, so I is a good set at every step of the algorithm, as claimed. □

6.3.2 Matroid Polytope

We now consider an integer programming formulation for the feasible region of problem (6.14). We will let

$$x_u = \begin{cases} 1 \text{ if } u \in S \text{ is selected} \\ 0 \text{ otherwise.} \end{cases}$$

The integer programming formulation to select an independent set from a matroid is:

$$\begin{aligned} x(U) := \sum_{u \in U} x_u \leq r(U) \qquad & \forall\, U \subseteq S \\ x_s \in \{0,1\} \qquad & \forall\, s \in S \end{aligned} \tag{6.16}$$

This is a correct formulation because if U is not independent, then $r(U) < |U|$. So the first constraint reads

$$\sum_{u \in U} x_u \leq r(U) < |U|,$$

so we cannot pick all the elements of U. In the reverse direction, this is a valid inequality, since for all independent sets U, $|U| \leq r(U)$ by the definition of rank.

We now relax integrality to define the matroid polytope. Note $x_s \leq 1$ is implied by $x(U) \leq r(U)$ for $U \subseteq S$ with $|U| = 1$, so our formulation is

$$\begin{aligned} x(U) &\leq r(U) && \forall\, U \subseteq S \\ x_s &\geq 0 && \forall\, s \in S \end{aligned} \tag{6.17}$$

It turns out that formulation (6.17) is integral. We will prove this by showing that the system in (6.17) is TDI.

Lemma 6.4. *Let $M = (S, C)$ be a matroid, and $w : S \to \mathbb{Z}_+$ where $|S| = n$. Then there exists $U^1, U^2, ..., U^n \subseteq S$, and $\lambda_1, \lambda_2, ..., \lambda_n \in \mathbb{Z}_+$ such that*

1. $w = \sum_{i=1}^{n} \lambda_i x^{U^i}$;
2. $\max\{w(I) \mid I \in C\} = \sum_{i=1}^{n} \lambda_i r(U^i)$.

where $x^{U^i} \in \{0,1\}^n$, and if element s_j is contained in U^i, $x_j^{U^i} = 1$, and 0 otherwise.

Proof. WLOG, we sort the elements of S as $s_1, s_2, ..., s_n$ such that

$$w(s_1) \geq w(s_2) \geq ... \geq w(s_n).$$

Then we define

$$\begin{aligned} &U^i = \{s_1, ..., s_i\} \\ &\lambda_i = w(s_i) - w(s_{i+1}) \ \forall i = 1, ..., n-1, \\ &\lambda_n = w(s_n). \end{aligned}$$

1. It is easy to check $w = \sum_{i=1}^{n} \lambda_i x^{U^i}$ by plugging in λ_i, U^i and x^{U^i}.
2. We have shown that $\max\{w(I) \mid I \in C\}$ can be obtained by the greedy algorithm, and we know $I = \{s_i \mid r(U^i) > r(U^{i+1}), i = 1, 2, ..., n-1\}$, then

$$\begin{aligned} \max\{w(I) \mid I \in C\} &= \sum_{i=1}^{n} w(s_i)[r(U^i) - r(U^{i+1})] \\ &= \sum_{i=1}^{n-1} [w(s_i) - w(s_{i+1})] r(U^i) + w(s_n) r(U^n) \\ &= \sum_{i=1}^{n} \lambda_i r(U^i) \end{aligned}$$

□

Consider following two problem,

$$
\begin{array}{llll}
(P) & \max \sum_{s\in S} w_s x_s & \qquad (D) & \min \sum_{U} y^U r^U \\
s.t. & \sum_{s\in U} x_s \le r(U) \quad \forall U \subseteq S & \qquad s.t. & \sum_{U \text{ that contains } s} y^U \ge w_s \quad \forall s \\
 & x_s \ge 0 & & y^U \ge 0
\end{array}
$$

Theorem 6.7. *(P) is integral.*

Proof. For $w : S \to \mathbb{Z}$, we can update $w : S \to \mathbb{Z}_+$, and in this way, by choosing the x with minimum coordinate number, we can obtain the same objective value with (P). By Lemma 1, define $\hat{y}^U$ as

$$\hat{y}^U = \begin{cases} \lambda_i & \text{if } U = U^i \\ 0 & \text{Otherwise.} \end{cases}$$

If w_s's are integral, λ_i's are integral, so that $\hat{y}^U$'s are integral. Since $r(U)$'s are integral since they are rank of U's. In this way, (P) is TDI with integral right hand side $r(U)$. Hence, by Proposition 3, (P) is integral. □

6.4 Suggested exercises

1. Given a graph, let S be the set of all of edges and $\mathcal{C}$ be the collection of subsets of edges which do not contain a cycle. Show that $M = (S, \mathcal{C})$ is a matroid.

2. Suppose you have n tasks to complete in n days. Each task requires your attention for a full day. Task j comes with a deadline d_j, the last day by which the job should be completed, and a penalty p_j that you must pay if you do not complete each task by its assigned deadline. The problem is to find the order to perform your tasks in order to minimize the total penalty you must pay. Prove that this problem can be solve in polynomial time. [Hint: Call a subset X of the tasks *good* if there is a schedule in which every task in X is on time. Prove that the collection of *good* task sets form a matroid.]

3. (*) Prove that the maching polytope for a given a graph $G(V, E)$:

$$\begin{aligned} \sum_{e \in \delta(v)} x_e &\leq 1 \;\forall v \in V \\ \sum_{e \in E(U)} x_e &\leq \frac{|U|-1}{2} \;\forall U \subseteq V, |U| \text{ odd} \\ x_e &\geq 0 \;\forall e \in E, \end{aligned}$$

is TDI.

Chapter 7

Introduction to Cutting-plane Theory

7.1 Introduction

In this section we will present different methods to obtain cutting planes. We will sort these methods in four categories: geometric ideas, relaxation based cuts and other approaches.

7.1.1 Geometric Ideas

We are given a polyhedron $P = \{x : Ax \leq b\}$, and we are interested in approximating $conv(P \cap \mathbb{Z}^n)$. '

Chvatal-Gomory Cutting Planes (Also called Gomory Fractional Cut)

Suppose we are given a valid inequality $\alpha x \leq \beta$ for $P \cap \mathbb{Z}^n$. If $\alpha \in \mathbb{Z}^n$, then $\alpha^\top x \in \mathbb{Z}$ for all $x \in P \cap \mathbb{Z}^n$ and therefore

$$\sum_{j=1}^{n} \alpha_j x_j \leq \lfloor \beta \rfloor$$

is also a valid inequality for $P \cap \mathbb{Z}^n$. This class of cutting-planes is called as the Chvatal-Gomoty cuts.

Let us illustrate this with an example, by considering the following set

$$\begin{aligned} x_1 + x_2 &\leq 3 \\ 5x_1 - 3x_2 &\leq 3 \\ x_1, x_2 &\geq 0 \\ x_1, x_2 &\in \mathbb{Z} \end{aligned}$$

A valid inequality is given by $4x_1 + 3x_2 \leq 10.5$. From the previous proposition, $4x_1 + 3x_2 \leq \lfloor 10.5 \rfloor$ is also a valid inequality for the set. This inequality is plotted in Figure 7.1.1. Let us also introduce the concept of *CG closure*.

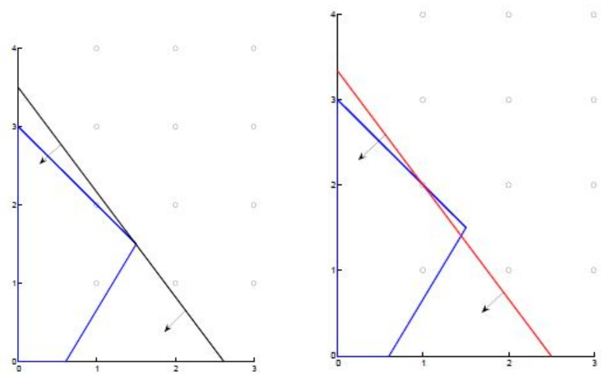


Figure 7.1: Chvatal-Gomory Cut

The idea is to measure the power of CG cuts by testing the power of all CG cuts added simultaneously.

Definition 7.1. *Let P be a polyhedron. The support function of P is denoted as follows:*

$$\sigma_P(\pi) := \max\{\pi^\top x \mid x \in P\}.$$

Note that all non-trivial CG cuts can then be seen to be of the following form:

$$\pi^\top x \leq \lfloor \sigma_P(\pi) \rfloor.$$

The CG closure is then defined as the set:

$$\bigcap_{\pi \in \mathbb{Z}^n \mid \sigma_P(\pi) < \infty} \{x \mid \pi^\top x \leq \lfloor \sigma_P(\pi) \rfloor.$$

We will study several properties of the CG closure.

Split Disjunctive Cuts

An arbitrary *split disjunction* is:

$$\pi^\top x \le \pi_0 \vee \pi^\top x \ge \pi_0 + 1 \text{ where } (\pi, \pi_0) \in \mathbb{Z}^n \times \mathbb{Z}.$$

Every integer point satisfies this disjunction, i.e., if $\hat{x} \in \mathbb{Z}^n$, then either $\pi^\top \hat{x} \le \pi_0$ or $\pi^\top \hat{x} \ge \pi_0 + 1$.

Let us consider a polyhedron P. We can now define two subsets:

$$P_1^\pi = \{x \in P : \pi^T x \le \pi_0\}$$
$$P_2^\pi = \{x \in P : \pi^T x \ge \pi_0 + 1\}$$

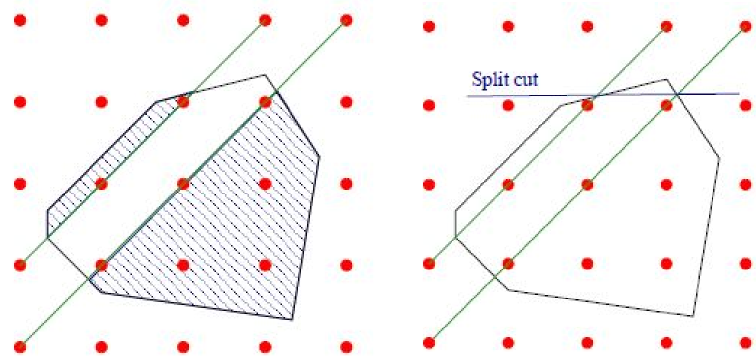
We then call $P^{\pi,\pi_0} := conv(P_1^\pi \cup P_2^\pi)$ the *disjunctive hull* with respect to the disjunction
$\pi^T x \le \pi_0 \vee \pi^T x \ge \pi_0 + 1$. The disjunctive hull is in fact a polyhedron. The valid inequalities for P^{π,π_0} can serve as cutting-planes. These are called *split cuts*.

Given a fixed disjunction $(\pi^\top x \le \pi_0) \vee (\pi^\top x \ge \pi_0 + 1)$ and fractional point $x^* \in P \setminus \mathbb{Z}^n$, we may ask does there exists a split cut that cuts off x^* from P^I. Let $P := \{x \,|\, Ax \le b\}$. It is easily verified that we can answer this question by solving the following LP, which essentially tries to find an inequality $\alpha^\top x \le \beta$ that is valid for both $P \cap \{x \,|\, \pi^\top x \le \pi_0\}$ and $P \cap \{x \,|\, \pi^\top x \ge 1\pi_0 + 1\}$ while simultaneosly attempting to separate x^* (using Farkas Lemma):

$$\begin{array}{lll} z^{CGLP}: & \max_{\alpha,\beta,\lambda^1,\lambda_0^1,\lambda^2,\lambda_0^2} & \alpha^\top x^* - \beta \\ & \text{s.t.} & \alpha^T = (\lambda^1)^\top A + \lambda_0^1 \pi^\top \\ & & \beta \ge (\lambda^1)^\top b + \lambda_0^1 \pi_0 \\ & & \alpha^T = (\lambda^2)^\top A + \lambda_0^2(-\pi^\top) \\ & & \beta \ge (\lambda^2)^\top b + \lambda_0^2(-\pi_0 - 1) \\ & & \lambda^1, \lambda^2, \lambda_0^1, \lambda_0^2 \ge 0 \\ & & \|\alpha\|_\infty \le 1, |\beta| \le 1 \quad \text{(Normalization)} \end{array}$$

where if $z^{CGLP} \le 0$ imples that we cannot separate the point x^* using the given disjunction or other wise $\alpha^\top x \le \beta$ is required inequality. The above LP is called the *cut-generating LP*.

Let us also introduce the concept of *split closure*. The idea is to look at the intersection of disjunctive hulls over all split disjunctions, and then study the resulting object. More precisely the split closure is defined as:



$$SC = \bigcap_{(\pi,\pi_0)\in\mathbb{Z}^n\times\mathbb{Z}} P^{\pi,\pi_0}$$

Notice that the split disjunctive cuts are a generalization of CG cuts, since each CG cut is obtained as a split cut when either $P_1^\pi = \emptyset$ or $P_2^\pi = \emptyset$.

Cutting-planes from lattice-free convex sets

One way to think about the split cut is that the set $S^{\pi,\pi} := \{x \mid \pi^0 \leq \pi^T x \leq pi^0 + 1\}$ contains no integer point in its interior. Then given a polyhedron P, we obtain cutting-planes by using valid inequalities for $P \setminus \text{int}(S^{\pi,\pi})$.

One may generalize this idea in the following fashion.

Definition 7.2 (Lattice-free convex set)**.** *A convex set Q is a lattice-free convex set if* $\text{int}(Q) \cap \mathbb{Z}^n = \emptyset$. *(Note integer points are allowed on the boundary).*

Now one uses valid inequalities for $P \setminus \text{int}(Q)$ to obtain cutting-planes.

Clearly, we would like to use large lattice-free sets to obtain cuts. This motivates the following definition.

Definition 7.3 (Maximal Lattice-free convex set)**.** *A set $S \subseteq \mathbb{R}^n$ is called a maximal lattice-free convex set if:*

1. *S is a lattice-free convex set*
2. *If $\tilde{S} \subseteq \mathbb{R}^n$ is a lattice-free convex set and $\tilde{S} \supseteq S$, then $\tilde{S} = S$.*

Maximal lattice-free convex set have some very nice properties.

Theorem 7.1. *Let $S \subseteq \mathbb{R}^n$ is a full-dimensional maximal lattice-free convex set. Then:*

1. *S is a polyhedron.*

2. *Every facet of S contains an integer point in its relative interior.*

3. *S has at most 2^n facets.*

While we do not prove Theorem 7.1 here, we present a proof sketch for the first part of Theorem7.1 under an additional boundedness assumption.

Proposition 7.1. *Let $S \subseteq \mathbb{R}^n$ is a full-dimensional maximal lattice-free convex set that is bounded. Then S is a polytope.*

Proof. Since S is bounded, we have that $S \subseteq \text{box}(M)$ where $\text{box}(M) := \{x \mid -M \leq x_j \leq M \;\forall j \in [n]\}$. Consider any integer point in $\text{box}(M)$. Since this integer point does not belong to $\text{int}(S)$ (which is a convex set), there exists a separating hyperplane that separates the integer point from $\text{int}(S)$. Since there is a finite number of integer points in $\text{box}(M)$, we obtain a finite number of separating hyperplanes. Consider the polytope defined by the separating hyperplanes. This polytope does not contain any integer point in its interior (by construction) and contains S. Thus, S is this polytope. $\square$

If $Q := \{x \mid (a^i)^T x \leq b^i, i \in \{1, \ldots, m\}\}$ is a lattice-free polyhedron, then we have that

$$\mathbb{Z}^n \subseteq \bigvee_{i=1}^{m} \{x | (a^i)^T x \geq b^i\}.$$

Thus, this is a mult term *disjunction* and therefore one may find the cutting planes for P if one uses valid inequalities for

$$\bigvee_{i=1}^{m} P \cap \{x | (a^i)^T x \geq b^i\}.$$

We can essentially set up a cut-generating LP again.

Cuts from Multiple simulataneous Splits

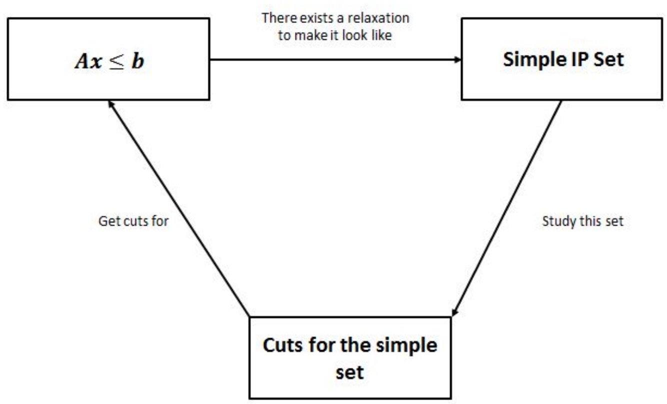
It is also possible to do multiple splits simultaneously as shown in the Figure below. We find valid inequalities for the resulting set, i.e. for $(\pi, \pi^0), (\eta, \eta^0) \in \mathbb{Z}^n \times \mathbb{Z}$ valid inequality for $P \cap \mathbb{Z}^n$ may be found by using valid inequalities for:

$$\begin{aligned}&\left(P \cap \{x \,|\, \pi^T x \le \pi^0\} \cap \{x \,|\, \eta^T x \le \eta^0\}\right)\\ \bigvee &\left(P \cap \{x \,|\, \pi^T x \le \pi^0\} \cap \{x \,|\, \eta^T x \ge \eta^0 + 1\}\right)\\ \bigvee &\left(P \cap \{x \,|\, \pi^T x \ge \pi^0 + 1\} \cap \{x \,|\, \eta^T x \le \eta^0\}\right)\\ \bigvee &\left(P \cap \{x \,|\, \pi^T x \ge \pi^0 + 1\} \cap \{x \,|\, \eta^T x \ge \eta^0 + 1\}\right).\end{aligned}$$

We can essentially set up a cut-generating LP again.

7.1.2 Cuts from relaxation

The basic idea of these cutting planes is illustrated in the Figure below:



Mixed Integer Rounding (Nemhauser and Wolsey)

Let us consider the set $X = \{(x, y) \in \mathbb{Z} \times \mathbb{R}_+ : x \le b + y\}$. Then

$$x \le \lfloor b \rfloor + \tfrac{1}{1-f} x$$

where $f = b - \lfloor b \rfloor$ is the fractional part of b, is a valid inequality for X. See Figure 7.1.2.

We can write “an extension” of the MIR for the following set

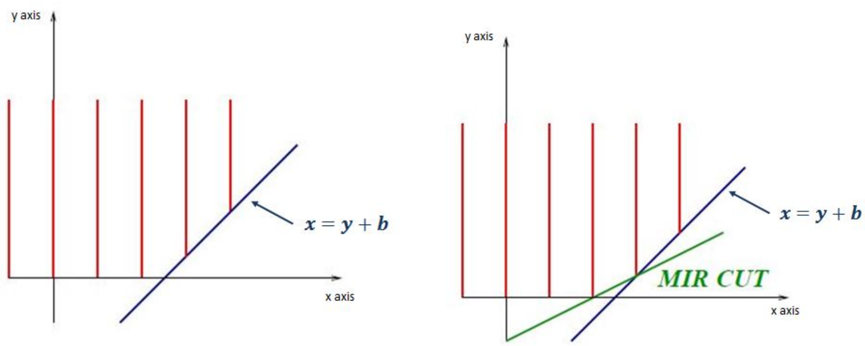


Figure 7.2: MIR

$$X' = \{(x, y) \in \mathbb{Z}^{|N|} \times \mathbb{R}_+ : \sum_{j \in N} a_j x_j + y^+ \leq b + y^-\}$$

to obtain the resulting cut:

$$\sum_{j \in N} \left(\lfloor a_j \rfloor + \tfrac{(f_j - f)^+}{1-f} \right) x_j \leq \lfloor b \rfloor + \tfrac{y^-}{1-f}$$

where $f_j = a_j - \lfloor a_j \rfloor$, that is a valid inequality for X'. The key observation is that X' is a one-row relaxation of any general mixed integer program, where all the continuous variables have been aggregated into two variables (one with positive coefficients, one with negative coefficients).

Knapsack (or Cover) Cuts

Let consider a knapsack type constraint in the form $\alpha^T x \leq \alpha_0$ where $\alpha \in \mathbb{Z}_+^{|V|}$, $\alpha_0 \in \mathbb{Z}_+$ and V is a subset of indices of binary variables x. A set $Q \subseteq V$ is called a *cover* if $\sum_{j \in Q} \alpha_j > \alpha_0$. In other words, Q is a set of binary variables which cannot be all together non-zero at the same time. In the light of this definition, the simplest version of a cover inequality is

$$\sum_{j \in Q} x_j \leq |Q| - 1$$

For instance, let us consider the following inequality

$$\begin{aligned} 3x_1 + 5x_2 + 4x_3 + 2x_4 + 7x_5 &\leq 8 \\ 0 \leq x_1, x_2, x_3, x_4, x_5 &\leq 1 \\ x_1, x_2, x_3, x_4, x_5 &\in \mathbb{Z}_+ \end{aligned}$$

We have $5+4>8$, so x_2 and x_3 cannot be simultaneously equal to 1. Thus,

$$x_2 + x_3 \leq 1$$

defines a cover inequality.
Other possible cover inequalities are:

$$\begin{aligned} x_4 + x_5 &\leq 1 \\ x_1 + x_2 + x_3 &\leq 2 \\ x_1 + x_2 + x_4 &\leq 2 \\ x_1 + x_2 + x_5 &\leq 2 \\ &\vdots \end{aligned}$$

Flow Cover Cuts

"Flow Set": It consists in a flow problem with arcs incoming and outgoing to a single node: to each of these arcs is associated a continuous variables measuring the flow on the arc and upper bounded by the arc capacity, if the flow over the arc is non-zero then a binary variable associated with the same arc must be set to 1 (in order to model fixed charge). A flow balance on the node must also be satisfied.

It turns out for many problems such relaxations can be constructed. Then the resulting inequalities for this flow set is used as cuts.

7.1.3 Algebraic approach

Suppose $S = \mathbb{Z}^n \cap P$, where $P = \{x \in \mathbb{R}^n_+ : Ax \geq b\}$ (A is a $n \times m$ matrix). We want to give a functional description of valid inequalities for S.

Definition 7.4 (Subdditive Function)**.** *A function $\Phi : D \subseteq \mathbb{R}^m \to \mathbb{R}$ is called **subadditive** over D if*

$$\Phi(d_1) + \Phi(d_2) \geq \Phi(d_1 + d_2) \text{ for all } d_1, d_2, d_1 + d_2 \in D.$$

Definition 7.5 (Superadditive Function)**.** *A function $\Phi : D \subseteq \mathbb{R}^m \to \mathbb{R}$ is called **superadditive** over D if*

$$\Phi(d_1) + \Phi(d_2) \leq \Phi(d_1 + d_2) \text{ for all } d_1, d_2, d_1 + d_2 \in D.$$

Definition 7.6 (Non-Decreasing Function)**.** *A function $\Phi : D \subseteq \mathbb{R}^m \to \mathbb{R}$ is called **non-decreasing** over D if $d_1, d_2 \in D$, $d_1 - d_2$ non-negative vector implies $\Phi(d_1) - \Phi(d_2)$ non-negative vector.*

Proposition 7.2. *If $\Phi : \mathbb{R}^m \to \mathbb{R}$ is superadditive, non-decreasing and $\Phi(0) = 0$ then*

$$\sum_{j=1}^{n} \Phi(A^j)x_j \geq \Phi(b)$$

is a valid inequality for $S = \mathbb{Z}^n \cap \{x \in \mathbb{R}^n_+ : Ax \geq b\}$ for any (A, b).

Proof. We have to show the three following inequalities for all $x \in S$ to prove that the inequality in the proposition is valid for S:

1. $\sum_{j=1}^{n} \Phi(A^j)x_j \geq \sum_{j=1}^{n} \Phi(A^j x_j)$
2. $\sum_{j=1}^{n} \Phi(A^j x_j) \geq \Phi(Ax)$
3. $\Phi(Ax) \geq \Phi(b)$

Let us prove the different points using the hypotheses:

1. It suffices to show that $\Phi(A^j)x_j \geq \Phi(A^j x_j)$ for all j. We will prove it by induction. If $x_j = 0$, then $\Phi(A^j)x_j = 0 = \Phi(0) = \Phi(A^j x_j)$. If $x_j = 1$, then $\Phi(A^j)x_j = \Phi(A^j) = \Phi(A^j x_j)$. Suppose it is true for $x_j = k - 1$. Then:

$$\begin{aligned}
k\Phi(A^j) &= \Phi(A^j) + (k-1)\Phi(A^j) \\
&\geq \Phi(A^j) + \Phi((k-1)A^j) \text{ by induction hypothesis} \\
&\geq \Phi(A^j + (k-1)A^j) \text{ by subadditivity of the function} \\
&= \Phi(kA^j)
\end{aligned}$$

2. We have the following:

$$\begin{aligned}
\sum_{j=1}^{n} \Phi(A^j x_j) &= (\Phi(A^1 x_1) + \Phi(A^2 x_2)) + \sum_{j=3}^{n} \Phi(A^j x_j) \\
&\geq (\Phi(A^1 x_1 + A^2 x_2)) + \sum_{j=3}^{n} \Phi(A^j x_j) \text{ by subadditivity of the function} \\
&\geq \cdots \geq \Phi(Ax)
\end{aligned}$$

3. Since $Ax \geq b$ for all $x \in S$ and Φ non-decreasing, the inequality holds.

□

Proposition 7.3. *If* $\Phi : \mathbb{R}^m \to \mathbb{R}$ *is superadditive, non-decreasing and* $\Phi(0) = 0$ *then*

$$\sum_{j=1}^{n} \Phi(A^j)x_j \leq \Phi(b)$$

is a valid inequality for $S = \mathbb{Z}^n \cap \{x \in \mathbb{R}^n_+ : Ax \leq b\}$ *for any* (A, b).

Proof. Similar to the proof of the previous proposition. □

To illustrate this proposition, let consider the following system of inequalities

$$\begin{aligned} x_1 + x_2 \quad\quad &\leq 1 \\ x_2 + x_3 &\leq 1 \\ x_1 \quad\quad + x_3 &\leq 1 \end{aligned}$$

and suppose we can write it as $\{x \in \mathbb{R}^3_+ : Ax \leq b\}$.

Then,

$$x_1 + x_2 + x_3 \leq 1$$

is a valid inequality for $S = \mathbb{Z}^3 \cap \{x \in \mathbb{R}^3_+ : Ax \leq b\}$.

Indeed, if we consider the function $\Phi : \mathbb{R}^3 \to \mathbb{R}$ such that $\Phi(x_1, x_2, x_3) = \lfloor \frac{1}{2}(x_1+x_2+x_3) \rfloor$, we can easily check that it is superadditive, non-decreasing and $\Phi(0) = 0$. From the proposition,

$$\sum_{j=1}^{3} \Phi(A^j)x_j \leq \Phi(b)$$

that can be written as

$$\lfloor 1 \rfloor \cdot x_1 + \lfloor 1 \rfloor \cdot x_2 + \lfloor 1 \rfloor \cdot x_3 \leq \lfloor \tfrac{3}{2} \rfloor$$

is a valid inequality for $S = \mathbb{Z}^3 \cap \{x \in \mathbb{R}^3_+ : Ax \leq b\}$, hence the result.

7.1.4 Reformulation-Linearization Technique (RLT)

The RLT is designed for 0-1 linear programs. Consider the 0-1 IP feaible region: $P \cap \{0,1\}^n$. Note that the condition $x_i \in \{0,1\}$ may be re-written as

$$x_i^2 = x_i.$$

The steps are the following:

1. Reformulation step: Multiply $x_1 \geq 0$ and $1 - x_1 \geq 0$ to the constraints $Ax \leq b$:

$$\sum_{j=1}^{n} A_{ij}x_1x_j \leq b_i x_1, \quad \sum_{j=1}^{n} A_{ij}x_j - \sum_{j=1}^{n} A_{ij}x_1x_j \leq b_i - b_i x_1 \quad \forall i \in [m]$$

2. Linearization step: In the above system, replace x_1^2 by x_1 and replace x_1x_j by y_{1j} for $j \neq 1$, thus obtaining the linear system:

$$\begin{array}{rcl} A_{i1}x_1 + \sum_{j=2}^{n} A_{ij}y_{1j} & \leq & b_i x_1, \\ \sum_{j=1}^{n} A_{ij}x_j - A_{i1}x_1 - \sum_{j=2}^{n} A_{ij}y_{1j} & \leq & b_i - b_i x_1 \end{array} \quad \forall i \in [m]$$

3. Projection: Finally, we project the above polytope in the space of x and y variables to the spae of x variables. Let us call the resulting polytope Q^1. It is easily seen that

$$P \supseteq Q^1 \supseteq P^I.$$

4. Repeat recursively: If we take Q^1 and then apply the above process now with x_2 and $1 - x_2$. Let us call this polytope Q^2. We can then again repeat the same procedure with x_3 and so forth. Then it can be shown that:

$$P \supseteq Q^1 \supseteq Q^2 \supseteq Q^3 \supseteq Q^4 \cdots \supseteq Q^n = P^I.$$

7.2 Some Classical results on CG Cutting Planes

In this section we present some results on the CG cuts. These classical results are representative of the results that have also been obtained (to various degrees) for other classes of cuts discussed in the previous section. For simiplicity, we repeat the definition of CG closure.

Definition 7.7 (CG Closure)**.** *Let $P \subset \mathbb{R}^n$ be a rational polyhedron. Its Chvatal-Gomory (CG) closure $P^{(1)}$ is defined as*

$$P^{(1)} = \bigcap_{\substack{\alpha \in \mathbb{Z}^n \\ \sigma_P(\alpha) < \infty}} \{x \in \mathbb{R}^n | \alpha^\top x \leq \lfloor \sigma_P(\alpha) \rfloor\} \tag{7.1}$$

Inductively, define the k^{th} CG closure $P^k (k \geq 2)$ as

$$P^{(k)} = \bigcap_{\substack{\alpha \in \mathbb{Z}^n \\ \sigma_P^{(k-1)}(\alpha) < \infty}} \{x \in \mathbb{R}^n | \alpha^\top x \leq \lfloor \sigma_P^{(k-1)}(\alpha) \rfloor\} \tag{7.2}$$

Enevn though the CG closure $P^{(1)}$ is defined by an infinite number of hyper planes, it is surprisingly a polyhedron.

Theorem 7.2 (Schrijver 1980)**.** *Let P be a rational polyhedron, then P^1 is also a rational polyhedron.*

Proof. W.l.o.g, assume P is non empty and $P = \{x | Ax \leq b\}$ where A and b are integral. Any CG cut should be in the form $y^\top Ax \leq \lfloor y^\top b \rfloor$, where $y \in \mathbb{Q}^m$ and $y^\top A \in \mathbb{Z}^n$.

We claim there exists $w \in \mathbb{Q}^m$ such that $0 \leq w_i < 1$ and the CG cut $y^\top Ax \leq \lfloor y^\top b \rfloor$ is implied by the CG cut $w^\top Ax \leq \lfloor w^\top b \rfloor$. Indeed, define $w = y - \lfloor y \rfloor$, where $\lfloor . \rfloor$ is taken component wise. Then, $0 \leq w_i < 1$. Since, $w^\top A = y^\top A - \lfloor y \rfloor^\top A$ and the two terms in RHS are both integral, $w^\top A$ is integral. Since $\lfloor y \rfloor^\top b$ is integral, $\lfloor w^\top b \rfloor + \lfloor y \rfloor^\top b = \lfloor w^\top b + \lfloor y \rfloor^\top b \rfloor = \lfloor y^\top b \rfloor$. Now taking the combination of the two inequalities $w^\top A \leq \lfloor w^\top b \rfloor$ and $\lfloor y \rfloor^\top Ax \leq \lfloor y \rfloor^\top b$, we get $y^\top Ax \leq \lfloor y^\top b \rfloor$. Thus, $w^\top A \leq \lfloor w^\top b \rfloor$ implies the cut $y^\top Ax \leq \lfloor y^\top b \rfloor$.

Let $S = \{\alpha \,|\, \exists w, \text{s.t.} w^\top A = \alpha \in \mathbb{Z}^m,\ 0 \leq w_i < 1 \forall i\}$ is therefore the set of interesting left-hand-sides for enerating CG cuts. The set of feasible α's in S is finite which proves the result. □

Theorem 7.3. *Let P be a rational polyhedron. Then there exists a finite k such that $P^k = P_I$.*

In order to prove we will need a number of preliminary results in order to prove Theorem 7.3.

Lemma 7.1. *Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron. Let F be a face of P, then $F^{(1)} = P^{(1)} \cap F$.*

Proof. Since any CG cut of P is a CG cut of F, we have $F^{(1)} \subseteq P^{(1)}$. Therefore, $F^1 \subseteq P^1 \cap F$. To show the other direction we first make a claim.

Claim : If $c^\top x \le \lfloor \delta \rfloor$ is a CG cut for F, then there exists $c^* \in \mathbb{Z}^n$ and $\delta^* \in \mathbb{R}$ such that $(c^*)^\top x \le \lfloor \delta^* \rfloor$ is a CG cut for P and the set $\{x \,|\, (c^*)^\top x \le \lfloor \delta^* \rfloor\} \cap F \subset \{x \,|\, c^\top x \le \lfloor \delta \rfloor\} \cap F\}$. Indeed w.l.o.g, we set $F = \{x \,|\, A^1 x = b^1, A^2 x \le b^2\}$ and $P = \{x | A^1 x \le b^1, A^2 x \le b^2\}$. Since $c^\top x \le \delta$ is a supporting hyperplane of F, there exists y_1 (not necessarily non-negative) and $y_2 \ge 0$ such that $c = y_1^\top A^1 + y_2^\top A^2$ and $\delta = y_1^\top b^1 + y_2^\top b^2$. Define $(c^*)^\top = (y_1 - \lfloor y_1 \rfloor)^\top A^1 + y_2^\top A^2, \delta^* = (y_1 - \lfloor y_1 \rfloor)^\top b^1 + y_2^\top b^2$. By integrality of $\lfloor y_1 \rfloor^\top b, \lfloor \delta^* \rfloor + \lfloor y_1 \rfloor^T b = \lfloor \delta^* + \lfloor y_1 \rfloor^T b \rfloor = \lfloor \delta \rfloor$. Aggregating two valid inequalities $(c^*)^\top \le \lfloor \delta^* \rfloor$ and $\lfloor y_1 \rfloor^\top A^1 x = \lfloor y_1 \rfloor^\top b^1$ (an equality valid for F), we get $c^\top x \le \lfloor \delta \rfloor$. This proves the claim.

As a consequence of the above claim and Theorem 7.2, observe that:

$$
\begin{aligned}
F^1 = \bigcap_{i \in \text{finite set}} \{x \,|\, (c^i)^\top x \le \lfloor \delta^i \rfloor\} &= F \bigcap_{i \in \text{finite set}} \{x \,|\, (c^i)^\top x \le \lfloor \delta^i \rfloor\} \\
&\supseteq \bigcap_{i \in \text{finite set}} \{x \,|\, (c^{i*})^T x \le \lfloor \delta^{i*} \rfloor\} \\
&\supseteq F \cap P^1,
\end{aligned}
$$

where $(c^{i*})^\top x \le \lfloor \delta^{i*} \rfloor$ is the CG cut of P that corresponds to CG cut $(c^i)^\top x \le \lfloor \delta^i \rfloor$ of F as in the claim above. Thus we complete the proof. □

We obtain the following Corollary of the above Lemma.

Corollary 7.1. *Let $P \subset \mathbb{R}^n$ be a rational polyhedron. Let F be a face of P and let k be natural number, then $F^{(k)} = P^{(k)} \cap F$.*

Lemma 7.2. *Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron. There exist rational matrix G and rational vector f s.t.*

- $P^I = \{x \,|\, Gx \le f\}$
- $max\{g_i^T x \,|\, x \in P\} < \infty$ *where g_i is a row of G*

Proof. We will consider two cases:

- Case 1: $P^I \neq \phi$

  Then by the fundamental theorem of IP we have that

  $$P_i = \{x \,|\, \hat{G}x \le \hat{f}\} \tag{7.3}$$

  We claim that $\hat{G}$ satisfies max $\{\bar{g}_i^T x \,|\, x \in P\}$. Assume contradiction $max\{\bar{g}_i^T x | x \in P\} = \infty \implies \exists v \in vec.cone(P)$, s.t. $\bar{g}^T v > 0$. By the fundamental theorem of integer programming, $v \in rec.cone(P^I) \implies \bar{g}_i^T x \le f_i$ cannot be a valid inequality for P^I.

- Case 2: $P^I = \phi$
  We claim that $dim(rec.cone(P)) \leq n-1$. Assume by contradiction $dim(rec.cone(P)) = n$. Let $y^1 \dots y^n$ be integral linearly independent vectors in $(rec.cone(P))$. Let $u \in \mathcal{Z}^n$. We can say that there exist λ_i s.t. $z + \sum_i \lambda_i y^i = u$. Consider $z + \sum_i (\lambda_i - \lfloor \lambda_i \rfloor) y^i \in P \cap \mathcal{Z}^n \implies P \cap Z^n \neq \phi$ which is a contradiction. From the claim above $\exists g \in \mathcal{Q}^n$ s.t. $g^T x = 0 \forall x \in rec.cone(P)$, which implies

$$max\{g^T x | x \in P\} < \infty \tag{7.4}$$

$$max\{-g^T x | x \in P\} < \infty \tag{7.5}$$

  Define $P^I = \{x | g^T x \leq -1, g^T x \geq 1\}$

□

The next result is related to the so-called Hermite normal form of a rational matrix. We will skip the proof of this result.

Lemma 7.3. *Let A be a full rank rational matrix. Then there exists a square invertible unimodular matrix U s.t. $AU = [B\ 0]$, where B is invertible.*

Lemma 7.4. *Let $P = \{x \,|\, Ax \leq b\}$ be a rational polyhedron. Let $Q = \{x \,|\, AUx \leq b\}$ where U is a unimodular matrix. Then $P^1 = \{Ax \leq b, CAx \leq \lfloor Cb \rfloor\} \Leftrightarrow Q^1 = \{AUx \leq b, CAUx \leq \lfloor Cb \rfloor\}$. Moreover $P^k = P^I \Leftrightarrow Q^k = Q^I$*

Proof. Since the inverse of a unimodular matrix is also a unimodular matrix, we only need to prove one direction. Clearly, $Q^1 \subseteq \{x | AUx \leq b, CAUx \leq \lfloor Cb \rfloor\}$. We need to prove that $Q^1 \supseteq \{AUx \leq b, CAUx \leq \lfloor Cb \rfloor\} = W$. Assume by contradiction $Q^1 \subsetneq W$. Suppose the CG cut is $(\lambda)^T AUx \leq \lfloor \lambda^T b \rfloor$ where $\lambda^T AU \in \mathbb{Z}^n$. Then observe that we have

$$max\{\lambda^T AUx | x \in W\} > \lfloor \lambda^T b \rfloor \tag{7.6}$$

By Farkas' lemma applied to W we have:

$$\alpha^T AU + B^T CAU = \lambda^T AU \tag{7.7}$$

$$\alpha^T b + B^T \lfloor Cb \rfloor > \lfloor \lambda^T b \rfloor \tag{7.8}$$

However this implies that $P^1 \neq \{Ax \leq b, CAx \leq \lfloor Cb \rfloor\}$ a contradiction. By induction on Part 1 we know $P^k = \{x | \tilde{A}x \leq \tilde{b}\} \Leftrightarrow Q^k = \{x | \tilde{A}Ux \leq \tilde{b}\}$.

To prove the second part we need to show that $\{x | \tilde{A}x \leq \tilde{b}\}$ is integral $\Leftrightarrow$ $\{y | \tilde{A}Uy \leq \tilde{b}\}$ is integral. Let F be a minimal face of P^k and $F =$

$\{x|A\bar{x} = \bar{b}\}$. The corresponding minimal face in Q is $\{x|\bar{A}Ux = \bar{b}\}$. Suppose $\hat{x} \in \mathbb{Z}^n$ and $\bar{A}\hat{x} = \bar{b}$. We know that $U^{-1}\hat{x} \in \mathbb{Z}^n$ and therefore we can say that $\bar{A}U(U^{-1}\hat{x}) = \bar{b}$. This means that if a minimal face of P^k contains an integer point, then an integer point is contained in the corresponding minimal face of Q^k as well. Since Q is obtained by a unimodular transformation of P, we can use the same technique to prove the converse. □

Finally, we are ready to prove Theorem 7.3.

Proof. of Theorem 7.3 The proof is based on induction of the dimension of the polyhedron P. Let $P \subseteq \mathbb{R}^n$ and $\dim(P) = d$.

Base case: When $d = 0$, P is a point in $\mathbb{R}^n$. If P is an integral point, then $P = P^I$; If P is not integral, assume $x_i^P \notin \mathbb{Z}$, where x_i^P is a component of P. In this case, $x_i \leq x_i^P$ is a valid inequality for P, and the corresponding CG cut is $x_i \leq \lfloor x_i^P \rfloor$, thus $P^{(1)} = \emptyset = P^I$. Therefore, the theorem holds when $d = 0$.

Inductive step: Suppose the theorem holds for all rational polyhedra with dimension $1, 2, \ldots, d-1$. Consider the following two cases:

- $\text{aff}(P) \cap \mathbb{Z}^n = \emptyset$. The affine hull of P can be written as $\text{aff}(P) = \{x \,|\, Cx = d\}$. Integer Farkas Lemma (Theorem 5.4) tells that there exists y $\in \mathbb{Q}^{m\times 1}$ such that $y^TC \in \mathbb{Z}^{1\times n}, y^Td \notin \mathbb{Z}$. Since $y^TCx \leq y^Td$ is a valid inequality for P, $y^TCx \leq \lfloor y^Td \rfloor$ is a CG cut. Denote it as (1); notice that $y^TCx \geq y^Td$ is also valid for P, thus $y^TCx \geq \lceil y^Td \rceil$ is also a CG cut. Denote it as (2). Combine (1) and (2) and it can be seen that $P^{(1)} = \emptyset$. Therefore, the theorem holds.

- $\text{aff}(P) \cap \mathbb{Z}^n \neq \emptyset$. Let z $\in \text{aff}(P) \cap \mathbb{Z}^n$ and consider the following polyhedron:
$$Q = \{u \,|\, u = v - z, v \in P\}.$$
It is sufficient to prove the theorem for Q (proof similar to that of Lemma 7.4). Let $Q = \{x : Ax \leq b\}$, and we may assume that the affine hull of Q is a linear subspace, i.e.,
$$\text{aff(Q)} = \{x : Cx = 0\}$$
By Lemma 7.3, there exists an unimodular matrix U such that CU = $\begin{bmatrix} B & 0 \end{bmatrix}$. Consider the following polyhedron
$$\text{T} = \{x : AUx \leq b\}$$
By Lemma 7.4, it is sufficient to prove the theorem for T. In addition,
$$\text{aff}(T) = \{x : CUx = 0\} = \{x : Bx^1 + 0x^2 = 0\}$$

Since B is nonsingular, $x^1 = 0$. Therefore, we may assume $T \subseteq \mathbb{R}^d$ and is full-dimensional. By Lemma 7.2, let $T^I = \{x : Gx \leq f\}$, then $\max\{g_1^\top x : x \in T\} = w < \infty$. Since T^I has a finite number of facet-defining inequalities, it is sufficient to prove that there exists $k < \infty$ such that $g_1^T x \leq f_1$ is a valid inequality for T^k.

Assume by contradiction, the statement above does not hold. Then there exists $f_1 < l \leq w$ and integer r such that $g_1^\top x \leq l$ is a valid face-defining inequality for $T^k, k \geq r$, and $g_1^\top x \leq l - 1$ is not a valid face-defining inequality for $T^k, k \geq r$. Define $F = T^r \cap \{x : g_1^T x = l\}$. Since F is a proper face of T^r, $\dim(F) \leq d - 1$ and $F \cap \mathbb{Z}^n = \emptyset$, otherwise $g_1^T x \leq f1$ cannot be a valid inequality for T^I.

By induction hypothesis, $F^\alpha = F^I = \emptyset$. By Corollary 7.1, $F^\alpha = T^{r+\alpha} \cap F = \emptyset$. This indicates that $\exists \epsilon > 0$ such that $g_1^T x \leq l - \epsilon$ is a valid inequality for $T^{r+\alpha}$. Then $g_1^T x \leq l - 1$ is a valid inequality for $T^{r+\alpha+1}$, which contradicts to the statement that $g_1^T x \leq l - 1$ is not a valid face-defining inequality for $T^k, k \geq r$. This completes the proof.

□

7.3 Split Cuts for 0-1 IPs

Theorem 7.3 shows that the CG rank is finite for rational polyhedron. It can be shown that for $0-1$ MILPs, the CG rank is atmost $\mathcal{O}(n^2 logn)$ where there are n binary variables. We show in this split cuts are stronger in the sense that the split rank for $0-1$ MILPs with n binary variables is at most n. We first need a preliminary result.

Lemma 7.5. *Let $S \subseteq \mathbb{R}^n$ and $a^T x \leq b$ be a valid inequality for S, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Then $conv(S) \cap H = conv(S \cap H)$, where $H = \{x : a^T x = b\}$.*

Proof. ($\supseteq$) Notice that $S \cap H \subseteq conv(S)$ and $S \cap H \subseteq H$. As $conv(S)$ and H are convex, we have $conv(S \cap H) \subseteq conv(S) \cap H$
($\subseteq$) Let $x \in conv(S) \cap H$. Then $x \in H$ and $x = \sum_{i \in I} \lambda_i x^i$, where $\sum_{i \in I} \lambda_i = 1, \lambda_i \geq 0, x^i \in S, \forall i$ in some index set I. Then

$$a^T x = \sum_i \lambda_i a^T x^i \leq \sum_i \lambda_i b = b$$

Since $x \in H$, we have $a^T x = b$ which implies that $a^T x^i = b$ for $\forall i \in I, i.e., x_i \in H$. Therefore, $x \in conv(S \cap H)$ □

Theorem 7.4. *Let* $P \subseteq [0,1]^n$ *and* $(P)_i = conv((P \cap \{x|x_i \leq 0\}) \cup (P \cap \{x|x_i \geq 1\}))$. *Then*

$$(..(((P)_1)_2)_3..)_n = P^I = T_n$$

where $S_i = P \cap \{x|x_j \in \{0,1\}, \forall j \in [i]\}, T_i = conv(S_i)$.

Proof. (By induction on n) Base case: $(P)_1 = T_1$
Inductive Step: Suppose for $k \geq 2$, it holds that $(..(((P)_1)_2)_3..)_{k-1} = T_{k-1}$. Then

$$\begin{aligned}(..(((P)_1)_2)_3..)_k &= conv[(T_{k-1} \cap \{x|x_k \leq 0\}) \cup (T_{k-1} \cap \{x|x_k \geq 1\})] \\ &= conv[(conv(S_{k-1}) \cap \{x|x_k = 0\}) \cup (conv(S_{k-1}) \cap \{x|x_k = 1\})] \\ &= conv[conv(S_{k-1} \cap \{x|x_k = 0\}) \cup conv(S_{k-1} \cap \{x|x_k = 1\})], \text{ by Lemma 1} \\ &= conv[(S_{k-1} \cap \{x|x_k = 0\}) \cup (S_{k-1} \cap \{x|x_k = 1\})] \\ &= conv(S_k) \\ &= T_k\end{aligned}$$

□

7.4 MIR, Gomory mixed integer (GMI) cut, Split cuts

Miixed integer rounding (MIR) is a cut for one-row relaxation of mixed integer sets, GMI cuts is algebraic cut, and Split cuts is a geometric cut. In this section, we show that these three classical cutting-planes are essentially the same.

7.4.1 MIR is split cut

We will work with a slightly different way to write the MIR set here. Consider the set:

$$x + y \geq b, x \in \mathbb{Z}, y \geq 0. \tag{7.9}$$

A valid inequality for this set is

$$x + \frac{1}{f}y \geq \lceil b \rceil, \tag{7.10}$$

where $f = b - \lfloor b \rfloor$.

Note now that this inequality can be obtained by appling a split disjunction:

Proposition 7.4. *The inequality (7.10) is valid for* $S^{\leq} := \{(x,y) \mid x+y \geq b, y \geq 0, x \leq \lfloor b \rfloor$ *and for* $S^{\geq} := \{(x,y) \mid x+y \geq b, y \geq 0, x \geq \lceil b \rceil\}$.

Proof.

- $S^{\leq}$: Observe that:

$$\begin{array}{rccc l} & x+y & \geq & b & \times\left(\frac{1}{f}\right) \\ + & -x & \geq & -\lfloor b \rfloor & \times\left(\frac{1-f}{f}\right) \\ \hline = & x+\frac{1}{f}y & \geq & \lceil b \rceil. & \end{array} \tag{7.11}$$

- $S^{\geq}$: Observe that:

$$\begin{array}{rccc l} & y & \geq & 0 & \times\left(\frac{1}{f}\right) \\ + & x & \geq & \lceil b \rceil & \times(1) \\ \hline = & x+\frac{1}{f}y & \geq & \lceil b \rceil. & \end{array} \tag{7.12}$$

□

Note that the MIR inequality can be applied to the following set:

$$\sum_{j \in I} a_j x_j + \sum_{j \in C} g_j y_j \geq b \tag{7.13}$$

$$x_j \in \mathbb{Z}_+ j \in I, y_j \geq 0 \ \forall y \in C. \tag{7.14}$$

Let $f_j := a_j - \lfloor a_j \rfloor$ for $j \in I$ and $f = b - \lfloor b \rfloor$. Let $I^{\leq} := \{j \in I \mid f_j \leq f\}$ and $I^{\geq} := \{j \in I \mid f_j \geq f\}$, let $C^+ := \{j \in C \mid g_j \geq 0\}$ and let $C^- := \{j \in C \| g_g \leq 0\}$. Now to use MIR inequality "optimally", we re-arrange and relax (7.13) as follows:

$$\underbrace{\left(\sum_{j \in I^{\leq}} \lfloor a_j \rfloor x_j + \sum_{j \in I^{\geq}} \lceil a_j \rceil x_j\right)}_{\in \mathbb{Z}} + \underbrace{\left(\sum_{j \in I^{\leq}} (a_j - \lfloor a_j \rfloor) x_j + \sum_{j \in C^+} g_j y_j\right)}_{\geq 0} \geq b$$

to obtain the inequality:

$$\sum_{j \in I^{\leq}} \left(\lfloor a_j \rfloor + \frac{f_j}{f}\right) x_j + \sum_{j \in I^{\geq}} \lceil a_j \rceil x_j + \sum_{j \in C^+} \frac{g_j}{f} y_j \geq \lceil b \rceil. \tag{7.15}$$

Indeed, in this case it is easily verifyed that (7.15) can be obtained by appling the disjunction

$$\left(\sum_{j\in I^{\leq}} \lfloor a_j \rfloor x_j + \sum_{j\in I^{\geq}} \lceil a_j \rceil x_j\right) \leq \lfloor b \rfloor \bigvee \left(\sum_{j\in I^{\leq}} \lfloor a_j \rfloor x_j + \sum_{j\in I^{\geq}} \lceil a_j \rceil x_j\right) \geq \lceil b \rceil$$

to the LP relaxation of (7.13).

7.4.2 A GMI cut is a MIR

The Gomory mixed integer cut is a cut of the algebraic type (Section 7.1.3). It is usually applied to rows of the simplex tableau corresponding to basic variables that are fractional. In particular, given the set

$$\sum_{i\in I} a_j x_j + \sum_{j\in C} g_j y_j = b,\ x_j \in \mathbb{Z}_+\ \forall j \in I,\ y_j \geq 0\ \forall j \in C. \tag{7.16}$$

The GMI cut is:

$$\begin{aligned}\sum_{j\in I, f_j\leq f} \frac{f_j}{f} x_j \quad &+ \sum_{j\in I, f_j\geq f} \frac{1-f_j}{1-f} x_j \\ &+ \sum_{j\in C, s.t. g_j\geq 0} \frac{1}{f} g_j y_j + \sum_{j\in C, s.t. g_j\leq 0} \frac{-g_j}{1-f} y_j \geq 1,\end{aligned} \tag{7.17}$$

where $f_j = a_j - \lfloor a_j \rfloor$.

Proposition 7.5. *The GMI cut above (valid for (7.16)) is implied by the MIR inequality applied to the following relaxation of (7.16):*

$$\sum_{i\in I} a_j x_j + \sum_{j\in C} g_j y_j \geq b,\ x_j \in \mathbb{Z}_+\ \forall j \in I,\ y_j \geq 0\ \forall j \in C. \tag{7.18}$$

Proof. Applying the MIR inequality to (7.18) we obtain

$$\sum_{j\in I, f_j\leq f} \left(\lfloor a_j \rfloor + \frac{f_r}{f}\right) x_j + \sum_{j\in I, f_j\geq f} \lceil a_j \rceil x_j + \sum_{j\in C, s.t. g_j\geq 0} \frac{g_j}{f} y_j \geq \lceil b \rceil. \tag{7.19}$$

Substracting (7.16) from (7.19) to obtain:

$$\begin{aligned}\sum_{j\in I, f_j\leq f} \left(-f_j + \frac{f_j}{f}\right) x_j \quad &+ \sum_{j\in I, f_j\geq f} (1-f_j) x_j \\ &+ \sum_{j\in C, g_j\leq 0} -g_j y_j + \sum_{j\in C, g_j\geq 0} \left(\frac{g_j}{f} - g_j\right) y_j \geq 1-f\end{aligned}$$

Diving the above inequality by $1-f$ yields the GMI cut. $\square$

7.4.3 A split cut is a GMI cut

In this section, we will show that any cut that is obtained as a split cut can also be obtained as GMI cut. It will be convenent to work with a slightly different version of GMI cut. In particular, we can re-write GMI cut (7.17) as:

$$\sum_{j\in J}\left(\lfloor a_j\rfloor+\frac{(f_j-f)^+}{1-f}\right)x_j+\frac{1}{1-f}\sum_{j\in C,s.t.g_j\le 0}g_jy_j\le\lfloor b\rfloor, \tag{7.20}$$

which is obtained as (7.16) - $f\times$(7.17), where $(t)^+=\max(t,0)$.

Before we prove the main result, we prsent a re-statement of Farkas' Lemma.

Lemma 7.6. *Let $Q=\{x\in\mathbb{R}: Ax\le b\}$ and $Q^{\le}=\{x\in P:\pi^\top x\le\pi_0\}$. If $Q^{\le}\ne\emptyset$ and $\alpha^\top x\le\beta$ is a valid inequality for $Q^{\le}$, then $\exists\mu\ge 0$ such that $\alpha^\top x-\mu\left(\pi^\top x-\pi_0\right)\le\beta$ is a valid inequality for Q.*

Proof. Since $\alpha^\top x\le\beta$ is a valid inequality for $Q^{\le}$, then by inhomogeneous Farkas Lemma, there exists $v\in\mathbb{R}^m_+,\mu\in\mathbb{R}_+$ such that $\alpha=A^\top v+\pi\mu,\beta\ge b^\top v+\pi_0\mu$. As $v\in\mathbb{R}^m_+$, $v^\top Ax\le v^\top b$ is a valid equality for Q. Meanwhile, as $A^\top v=\alpha-\pi\mu,\beta-\pi_0\mu\ge b^\top v$, we can get $(\alpha-\pi\mu)^\top x\le\beta-\pi_0\mu$ is also a valid inequality for Q; i.e., $\alpha^\top x-\mu\left(\pi^\top x-\pi_0\right)\le\beta$ is a valid inequality for Q. $\square$

Theorem 7.5. *Let $P:=\{(x,y)\in(\mathbb{R}^n_+\times\mathbb{R}^p_+)\mid Ax+Gy\le b\}$. If $c^\top x+d^\top y\le h$ is a valid split cut for $P\cap(\mathbb{Z}^n\times\mathbb{R}^p)$ obtained using a split disjunction: $\pi^\top x\le\pi_0\vee\pi^\top x\ge\pi+0+1$ and assuming that $P^{\pi,\pi_0}\ne\emptyset$, then $c^\top x+d^\top y\le h$ can be obtained as a GMI cut (i.e. we can obtain an implied equation for P by addition of slack variables, then applying the GMI cut to the implied equation, which yields $c^\top x+d^\top y\le h$).*

Proof. If $P=\emptyset$, the statement holds trivially. Thus, we now assume $P\ne\emptyset$. Let's define $P^{\le}:=P\cap\{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m:\pi^\top x\le\pi_0\},P^{\ge}:=P\cap\{(x,y)\in\mathbb{R}^n\times\mathbb{R}^m:\pi^\top x\ge\pi_0+1\}$. There are three different cases:

1. We first assume that $P^{\ge}\ne\emptyset,P^{\le}\ne\emptyset$. Since $c^\top x+d^\top y\le h$ is a valid inequality for $P^{\le}$ and also for $P^{\ge}$. By the previous lemma, there exists $\alpha\ge 0,\beta\ge 0$ such that $c^\top x+d^\top y-\alpha\left(\pi^\top x-\pi_0\right)\le h$ is a valid inequality for $P^{\le}$, $c^\top x+d^\top y+\beta\left(\pi^\top x-\pi_0-1\right)\le h$ is a valid inequality for $P^{\ge}$, which can be rewritten as

$$c^\top x+d^\top y-\alpha\left(\pi^\top x-\pi_0\right)+s_1=h, \tag{7.21}$$

$$c^\top x + d^\top y + \beta\left(\pi^\top x - \pi_0 - 1\right) + s_2 = h, \tag{7.22}$$

with $s_1 \geq 0, s_2 \geq 0$. We note that if $\alpha = 0$ or $\beta = 0$. There is nothing to prove since $c^\top x + d^\top y \leq h$ is a valid inequality for P.

Now suppose $\alpha > 0, \beta > 0$. Let (7.22) – (7.21), which yields

$$(\alpha + \beta)\,\pi^\top x + s_2 - s_1 = \beta + \pi_0\,(\alpha + \beta)$$

or equivalently,

$$\pi^\top x + \frac{s_2}{\alpha + \beta} - \frac{s_1}{\alpha + \beta} = \frac{\beta}{\alpha + \beta} + \pi_0.$$

Let $f_0 = \frac{\beta}{\alpha+\beta}$ and apply $GMIC$ procedure to the above equation (note that $(\pi, \pi_0) \in \mathbb{Z}^n \times \mathbb{Z}$). Then we arrive at

$$\pi^\top x - \frac{1}{1 - f_0}\frac{s_1}{\alpha + \beta} =\leq \pi_0;$$

i.e.,

$$\pi^\top x - \frac{s_1}{\alpha} \leq \pi_0.$$

Since $s_1 = c^\top x + d^\top y - \alpha\left(\pi^\top x - \pi_0\right) - h$, we can finally get

$$\pi^\top x - \frac{1}{\alpha}\left[c^\top x + d^\top y - \alpha\left(\pi^\top x - \pi_0\right) - h\right] \leq \pi_0,$$

i.e., $c^\top x + d^\top y \leq h$, which is $GMIC$ for P.

2. If $P^{\geq} = \emptyset$, which implies $P^{\leq} \neq \emptyset$ as $P^{\pi,\pi_0} \neq \emptyset$. As $P^{\leq} \neq \emptyset$, by the previous lemma, there exists $\alpha \geq 0$ such that $c^\top x + d^\top y - \alpha\left(\pi^\top x - \pi_0\right) \leq h$ is a valid inequality for $P^{\leq}$. Meanwhile, as $P^{\geq} = \emptyset$, $\max_{x \in P}\left(\pi^\top x - \pi_0 - 1\right) < -\delta, 1 > \delta > 0$. Choose $\beta = \frac{1-\delta}{\delta}\alpha \geq 0$. Then we have

$$\begin{aligned}
&c^\top x + d^\top y + \beta\left(\pi^\top x - \pi_0 - 1\right) = c^\top x + d^\top y + \frac{1-\delta}{\delta}\alpha\left(\pi^\top x - \pi_0 - 1\right) \\
&= c^\top x + d^\top y + \alpha\left[-\left(\pi^\top x - \pi_0\right) + 1 + \frac{1}{\delta}\left(\pi^\top x - \pi_0 - 1\right)\right] \\
&\leq c^\top x + d^\top y - \alpha\left(\pi^\top x - \pi_0\right) + \left(1 - \frac{1}{\delta}\delta\right) \leq h.
\end{aligned}$$

Thus, $c^\top x + d^\top y + \beta\left(\pi^\top x - \pi_0 - 1\right)$ is a valid inequality for P. Now the same procedure of case (i) follows.

3. If $P^{\leq} = \emptyset$, which implies $P^{\geq} \neq \emptyset$. As $P^{\geq} \neq \emptyset$, by the previous lemma, there exists $\beta \geq 0$ such that $c^\top x + d^\top y + \beta \left(\pi^\top x - \pi_0 - 1\right) \leq h$ is a valid inequality for $P^{\geq}$. Meanwhile, as $P^{\leq} = \emptyset$, $\min_{x \in P} \left(\pi^\top x - \pi_0\right) > \delta, 1 > \delta > 0$. Choose $\alpha = \frac{1-\delta}{\delta}\beta \geq 0$. Then we have

$$
\begin{aligned}
&c^\top x + d^\top y - \alpha \left(\pi^\top x - \pi_0\right) = c^\top x + d^\top y - \frac{1-\delta}{\delta}\beta \left(\pi^\top x - \pi_0\right) \\
&= c^\top x + d^\top y + \beta \left[\left(\pi^\top x - \pi_0 - 1\right) + 1 - \frac{1}{\delta}\left(\pi^\top x - \pi_0\right)\right] \\
&\leq c^\top x + d^\top y + \beta \left(\pi^\top x - \pi_0 - 1\right) + \left(1 - \frac{1}{\delta}\delta\right) \leq h.
\end{aligned}
$$

Thus, $c^\top x + d^\top y - \alpha \left(\pi^\top x - \pi_0\right)$ is a valid inequality for P. Now the same procedure of case (i) follows.

$\square$

7.5 Cover Inequality

7.5.1 Knapsack Cover Inequality

Let $K = \{x \in \{0,1\}^n | \sum_{j=1}^n a_j x_j \leq b\}$ be the feasible set of a Knapsack problem, where we can assume $0 \leq a_j \leq b$ for $j = 1, ..., n$. Then $C \subseteq [n]$ is a cover if $\sum_{j \in C} a_j > b$.

Observation 7.1. *If C is a cover, then $\sum_{j \in C} x_j \leq |C| - 1$ is a valid inequality for conv(K). A cover is minimal if $\sum_{j \in C/\{k\}} \leq b$ for $\forall k \in C$. For example, let $C = \{1,2,3\}, a_k = k + 2$ for $k \in C$ and $b = 11$. Then C is a minimal cover.*

Definition 7.8 (Cover inequality)**.** $\sum_{j \in C} x_j \leq |C| - 1$

Proposition 7.6. *If C is a minimal cover, then the cover inequality is facet-defining for $conv(K) \cap \{x | x_i = 0, \forall i \notin C\}$*

7.5.2 Flow Cover Inequality

Let $S = \{(x,y) \in \{0,1\}^n \times \mathbb{R}_+^n | \sum_{j=1}^n y_j \leq b, y_j \leq a_j x_j, \forall j \in [n]\}$ and let $T = \{(x,y) \in \{0,1\}^n \times \mathbb{R}_+ | \sum_{j=1}^n a_j x_j \leq b + y\}$.

Definition 7.9 (Flow Cover Inequality)**.** $\sum_{j \in C} y_j + \sum_{j \in C} (a_j - \lambda)^+ (1 - x_j) \leq b$*, where* $\lambda = \sum_{j \in C} a_j - b$

Proposition 7.7. *Let C be a cover and $\lambda = \sum_{j \in C} a_j - b$. Then the inequality*

$$\sum_{j \in C} \min\{a_j, \lambda\} x_j \le \sum_{j \in C} \min\{a_j, \lambda\} - \lambda + y$$

is a valid inequality for $\text{conv}(T)$.

Proof. Let $C^1 = \{j | a_j \le \lambda\}$, $C^2 = \{j | a_j > \lambda\}$, and $(x, y) \in T$.

- Case I: $x_j = 0$ for some $j \in C^2$
  Then

$$\begin{aligned} LHS &= \sum_{j \in C^1} a_j x_j + \sum_{j \in C^2} \lambda x_j \\ &\le \sum_{j \in C^1} a_j + \lambda(|C^2| - 1) \\ &\le \sum_{j \in C^1} a_j + \lambda |C^2| - \lambda + y. \end{aligned}$$

- Case 2: $x_j = 1$ for all $j \in C^2$.
  Let $C^{1+} = \{j | x_j = 1\}$ and $C^{1-} = \{x_j | x_j = 0\}$. Then

$$\begin{aligned} LHS &= \sum_{j \in C^1} a_j x_j + \sum_{j \in C^2} \lambda x_j \\ &= \sum_{j \in C^{1+}} a_j + \lambda |C^2| \\ &= \sum_{j \in C^1} a_j - \sum_{j \in C^{1-}} a_j + \lambda |C^2| \\ &\le \sum_{j \in C^1} a_j + y - \lambda + \lambda |C^2|, \end{aligned}$$

  where the last inequality follows from the definition of λ and T:

$$\begin{aligned} & \sum_{j \in C} a_j x_j \le b + y \\ \Longrightarrow & \sum_{j \in C} a_j - \sum_{j \in C^{1-}} a_j \le b + y, \text{ by definition of } C^{1-} \\ \Longrightarrow & \lambda - \sum_{j \in C^{1-}} a_j \le y \\ \Longrightarrow & -\sum_{j \in C^{1-}} a_j \le y - \lambda \end{aligned}$$

□

Observation 7.2. *We applied the disjunction* $\sum_{j\in C^2} x_j \leq |C^2| - 1$ *and* $\sum_{j\in C^2} x_j \geq |C^2|$.

We want to show that the following flow cover inequality

$$\sum_{j\in C} y_j + \sum_{j\in C} (a_j - \lambda)^+ \cdot (1 - x_j) \leq b. \tag{7.23}$$

is a valid inequality for $conv(S)$.

Proposition 7.8. *Inequality (7.23) is a v.i. for the convex hull of S.*

Proof. Let us re-write the constraints $y_j \leq a_j x_j, \forall j \in [n]$ into

$$y_j + t_j = a_j x_j,\ t_j \geq 0\ \forall j \in [n],. \tag{7.24}$$

Combining with the constraints $\sum_{j\in C} y_j \leq b$, we have

$$\sum_{j\in C} a_j x_j - \sum_{j\in C} t_j \leq b$$

$\Rightarrow$

$$\sum_{j\in C} a_j x_j \leq b + \sum_{j\in C} t_j.$$

By Proposition 7.7 in previous lecture, we have

$$\sum_{j\in C} \min\{a_j, \lambda\} \cdot x_j \leq \sum_{j\in C} \min\{a_j, \lambda\} - \lambda + \sum_{j\in C} t_j$$

$\Rightarrow$

$$\sum_{j\in C} \min\{a_j, \lambda\} \cdot x_j \leq \sum_{j\in C} \min\{a_j, \lambda\} - \sum_{j\in C} a_j + b + \sum_{j\in C} a_j x_j - \sum_{j\in C} y_j$$

$\Rightarrow$

$$\sum_{j\in C} y_j + \sum_{j\in C} (\min\{a_j, \lambda\} - a_j) \cdot x_j + \sum_{j\in C} (a_j - \min\{a_j, \lambda\}) \leq b$$

$\Rightarrow$

$$\sum_{j\in C} y_j + \sum_{j\in C} (a_j - \lambda)^+ \cdot (1 - x_j) \leq b.$$

□

We do not prove the next result.

Theorem 7.6. *Inequality (7.23) is facet-defining for conv(S) if* $\max_{j\in C} > \lambda$.

Example: Capacitated Facility Location Problem
Notations:

- y_{ij}: quantity from facility j to retailer i.
- $x_j = \begin{cases} 1 & \text{if facility } j \text{ is open} \\ 0 & \text{otherwise.} \end{cases}$
- u_j: capacity of facility j.
- d_i: demand of retailer i.

Constraints:

$$\begin{aligned} \sum_j y_{ij} &= d_i,\ \forall i \\ \sum_i y_{ij} &\leq u_j x_j,\ \forall j \\ y_{ij} &\geq 0\ \forall i,\ j \\ x_j &\in \{0,1\} \end{aligned}$$

We define $z_j = \sum_i y_{ij}\ \forall j$, and we have

$$z_j \leq u_j x_j,\ \forall j$$

and

$$\sum_j z_j = \sum_i d_i \quad \Rightarrow \quad \sum_j z_j \leq \sum_i d_i$$

Then we can use Proposition 7.8 to generate valid inequalities.

7.6 Lifting

Proposition 7.9. *Consider a set* $S \subseteq \{0,1\}^n$, *s.t.* $S \bigcap \{x | x_n = 1\} \neq \emptyset$. *Suppose* $\sum_{j=1}^{n-1} \alpha_j x_j \leq \beta$ *is valid for* $S \bigcap \{x | x_n = 0\}$. *Then*

$$\alpha_n = \beta - \max\{\sum_{j=1}^{n-1} \alpha_j x_j | x \in S,\ x_n = 1\}$$

is the largest coefficient s.t. $\sum_{j=1}^{n} \alpha_j x_j \leq \beta$ *is valid for* S. *Moreover, if* $\sum_{j=1}^{n-1} \alpha_j x_j \leq \beta$ *induces a d-dimensional face w.r.t.* $conv(S) \bigcap \{x | x_n = 0\}$, *then* $\sum_{j=1}^{n} \alpha_j x_j \leq \beta$ *induces a face of* $conv(S)$ *whose dimension is at least* $d+1$.

Proof. (i). To prove $\sum_{j=1}^{n} \alpha_j x_j \leq \beta$ is a v.i. for S. Let $\hat{x} \in S$. If $\hat{x}_n = 0$, then by hypothesis we have $\sum_{j=1}^{n-1} \alpha_j \hat{x}_j + \alpha_n \cdot 0 \leq \beta$. If $\hat{x}_n = 1$, by the definition of α_n, we have $\sum_{j=1}^{n-1} \alpha_j \hat{x}_j \leq \beta - \alpha_n$. Thus, $\sum_{j=1}^{n} \alpha_j \hat{x}_j \leq \beta - \alpha_n + \alpha_n \cdot 1 = \beta$.

(ii). Let $\bar{x}$ be an optimal solution to the maximization problem

$$\max\{\sum_{j=1}^{n-1} \alpha_j x_j | x \in S,\ x_n = 1\}.$$

By definition of α_n, we know that $\forall \delta > 0$,

$$\sum_{j=1}^{n-1} \alpha_j \bar{x}_j + (\alpha_n + \delta) \cdot 1 > \beta.$$

Hence, the largest coefficient is α_n.

(iii). By hypothesis, there exists $d+1$ affinely independent points in $S \bigcap \{x | x_n = 0\}$ satisfying the inequality $\sum_{j=1}^{n} \alpha_j x_j \leq \beta$ at equality. Now let us take the point $(\bar{x}, 1)$. Clearly, by construction of α_n, we have

$$\sum_{j=1}^{n-1} \alpha_j \bar{x}_j + \alpha_n \cdot 1 = \beta.$$

Furthermore, $(\bar{x}, 1)$ is affinely independent from the previous $d+1$ points. Therefore, $\sum_{j=1}^{n} \alpha_j x_j \leq \beta$ induces a face of *conv*(S) whose dimension is at least $d+1$. $\square$

Example: Consider the following system

$$8x_1 + 7x_2 + 6x_3 + 4x_4 + 6x_5 + 6x_6 + 6x_7 \leq 22.$$
$$x_j \in \{0, 1\},\ \forall j \in \{1, 2, ..., 7\}$$

Notice that $x_1 + x_2 + x_3 + x_4 \leq 3$ (which is a cover) is a v.i. Now let us lift x_5. Notice that the maximization problem

$$3 - \max\{x_1 + x_2 + x_3 + x_4 | 8x_1 + 7x_2 + 6x_3 + 4x_4 \leq 16\} = 1.$$

Therefore, we can obtained the strengthened (lifted) inequality

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 3.$$

Similarly, we can lift x_6 by

$$3 - \max\{x_1 + x_2 + x_3 + x_4 + x_5 | 8x_1 + 7x_2 + 6x_3 + 4x_4 + 6x_5 \leq 16\} = 0.$$

Notice that now the lifted inequality

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 3.$$

is exactly the original one. We call this process "*sequential lifting*". Note that we can lift x_7 as well and obtain the inequality

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 3.$$

Observations:

(1) Different lifting order may lead to different inequality.

(2) Sequential-lifting preserves the facet-structure of the inequality.

(3) Not all the facet-defining inequalities can be obtained by sequential-lifting.

The previous lifting-technique is called up-lifting, one can symmetrically construct the corresponding down-lifting technique. In particular, we have the following proposition.

Proposition 7.10. *Consider a set $S \subseteq \{0,1\}^n$, s.t. $S \bigcap \{x | x_n = 0\} \neq \emptyset$. Suppose $\sum_{j=1}^{n-1} \alpha_j x_j \leq \beta$ is valid for $S \bigcap \{x | x_n = 1\}$. Then*

$$\alpha_n = -\beta + \max\{\sum_{j=1}^{n-1} \alpha_j x_j | x \in S,\ x_n = 0\}$$

is the largest coefficient s.t. $\sum_{j=1}^{n} \alpha_j x_j \leq \beta + \alpha_n$ is valid for S. Moreover, if $\sum_{j=1}^{n-1} \alpha_j x_j \leq \beta$ induces a d-dimensional face w.r.t. conv$(S) \bigcap \{x | x_n = 1\}$, then $\sum_{j=1}^{n} \alpha_j x_j \leq \beta + \alpha_n$ induces a face of conv(S) whose dimension is at least $d + 1$.

A practical problem may arise when using the lifting-technique. That is, solving an IP to get a cut is too expensive. The following approach is a possible way out. Let $S = \{x \in \{0,1\}^n \mid Ax \leq b\}$ and $\sum_{j \in C} \alpha_j x_j \leq \beta$ be a valid inequality for $S \bigcap \{x | x_j = 0, \ \forall j \in [n] \setminus C\}$. Then consider the problem

$$\begin{aligned} f(z) : \quad = \quad & \min\{\beta - \sum_{j \in C} \alpha_j x_j\} \\ s.t. \quad & \sum_{j \in C} A^j x_j \leq b - z. \\ & x_j \in \{0,1\}, \forall j \in C. \end{aligned}$$

Theorem 7.7. *Let $g : \mathbb{R}^m \mapsto \mathbb{R}$ be a function that satisfies the following*

1. *g is superadditive, i.e., $g(u+v) \geq g(u) + g(v)$ for all u, v and $g(0) = 0$;*
2. *$g(x) \leq f(x)$ for all x.*

Then

$$\sum_{j \in C} \alpha_j x_j + \sum_{j \in [n] \setminus C} g(A^j) x_j \leq \beta$$

is a valid inequality for $\text{conv}(S)$*.*

We further introduce some notation. For simplicity we denote $g(A^j)$ also by α_j for $j \in [n] \setminus C$. Fix any arbitrary sequence of indices in $[n] \setminus C$, say $j_1, \ldots, j_{n-|C|}$. For $i = 0, \ldots, n - |C|$, we define $C_i = C \cup \{j_1, \ldots, j_i\}$ with $C_0 = C$. Further, let

$$\begin{aligned} [\Pi^i] \quad f_i(z) = \min \quad & \beta - \sum_{j \in C_{i-1}} \alpha_j x_j \\ \text{s.t.} \quad & \sum_{j \in C_{i-1}} A^j x_j \leq b - z \\ & x_j \in \{0,1\} \ \forall j \in C_{i-1}. \end{aligned}$$

We first prove the following lemma.

Lemma 7.7. *For all $i = 1, \ldots, |[n] \setminus C|$, $g(z) \leq f_i(z)$.*

Proof. We prove this by induction. When $i = 1$, $f_1 = f$, the result follows from the definition of g. Suppose the result is true for $1 \leq k \leq i-1$ with some $2 \leq i \leq n - |C| - 1$. Let x^* be the optimal solution to $[\Pi^i]$, and $u = A^{j_{i-1}} x^*_{j_{i-1}}$. Thus we have

$$\begin{aligned}
f_i(z) &= \beta - \sum_{j \in C_{i-2}} \alpha_j x^*_j - \alpha_{j_{i-1}} x^*_{j_{i-1}} \\
&= f_{i-1}(z+u) - g(A^{j_{i-1}}) x^*_{j_{i-1}} \text{ (by optimality of } x^*) \\
&\geq g(z+u) - g(A^{j_{i-1}}) x^*_{j_{i-1}} \text{ (by induction hypothesis)} \\
&= g(z+u) - g(A^{j_{i-1}} x^*_{j_{i-1}}) \text{ (since } x^*_{j_{i01}} \text{ are either 0 or 1)} \\
&= g(z+u) - g(u) \geq g(z). \text{ (by superadditivity of } g)
\end{aligned}$$
□

With this lemma, it is easy to prove Theorem 7.7.

Proof of Theorem 7.7. The idea is to lift variables $x_{j_1}, \ldots, x_{j_{n-|C|}}$ sequentially, but instead of using the largest possible coefficient $f_i(A^{j_i})$, we use $g(A^{j_i})$, which yields a weaker valid inequality. We explain the process for $i = 1$, the rest follows the same routine. Suppose we lift variable x_{j_1} and obtain is largest possible coefficient $f_1(A_{j_1})$, thus $\sum_{j \in C} \alpha_j x_j + f_1(A_{j_1}) x_{j_1} \leq \beta$ is a valid inequality for $\text{conv}(S) \cap \{x \,|\, x_j = 0 \;\forall j \in [n] \setminus C_1\}$. By Lemma 7.7, we know $f_1(A_{j_1}) \geq g(A_{j_1})$, thus $\sum_{j \in C} \alpha_j x_j + g(A_{j_1}) x_{j_1} \leq \beta$ is still a valid inequality for $\text{conv}(S) \cap \{x \,|\, x_j = 0 \;\forall j \in [n] \setminus C_1\}$. We continue to lift variable x_{j_2} and apply the same substitution. Finally after lifting all x variables, we obtain the desired valid inequality. □

Notice that Theorem 7.7 provides a more computationally efficient way to lift variables and generate valid inequalities, compared to solving problems like $[\Pi^i]$ to obtain the largest possible coefficients.

7.7 Suggested exercises

1. Construct an example of a polytope in $\mathbb{R}^2$ whose CG rank is great than 1.

2. Construct an example of a polytope in $\mathbb{R}^2$ whose split rank is great than 1.

3. Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron. For $\pi \in \mathbb{Z}^n$ and $\pi_0 \in \mathbb{Z}$, let $P^{\pi,\pi_0} := \text{conv}\{(P \cap \{x \mid \pi^\top x \leq \pi_0\}) \cup (P \cap \{x \mid \pi^x \geq \pi_0 + 1\})$. Recall that $\cap_{\pi \in \mathbb{Z}^n, \pi_0 \in \mathbb{Z}} P^{\pi,\pi_0}$ is the split closure of P.
   (a) For a fixed $\pi \in \mathbb{Z}^n$, let $P^\pi = \cap_{\pi_0 \in \mathbb{Z}} P^{\pi,\pi_0}$. Prove that $P^\pi = \{x \mid x \in P, \pi^T x \in \mathbb{Z}\}$.
   (b) Let A be a matrix with the following property: removing any one column from A makes the resulting matrix totally unimodular and let b be an integral vector. Prove that the split closure of $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ yields it integer hull (i.e. convex hull of integer feasible solutions)

4. Let A be a $m \times n$ rational matrix. Let c be a n dimensional rational vector. Consider the following function f:

$$\begin{aligned} f(u) = \quad & \max \quad c^T x \\ & \text{s.t.} \quad x \in S^u, \end{aligned}$$

   where $S^u := \{x \in \mathbb{Z}^n \mid Ax \leq u\}$ and f is defined over the set $S \subseteq \mathbb{R}^m$ where $S = \{u \in \mathbb{R}^m \mid S^u \neq \emptyset\}$. [1] Prove the following:
   (a) $f(0) = 0$ if and only if $f(u)$ is bounded for all $u \in S$.
   (b) f is non-decreasing, i.e., if $u^1 \geq u^2$ componentwise, then $f(u^1) \geq f(u^2)$.
   (c) f is superadditive, i.e., $f(u^1) + f(u^2) \leq f(u^1 + u^2)$.
   (d) Assume $f(0) = 0$. Show that there exists $g : \mathbb{R}^m \to \mathbb{R}$ such that
      i. $g(u)$ is finite valued for $u \in \mathbb{R}^m$,
      ii. $g(u) = f(u)\ \forall u \in S$,
      iii. g is non-decreasing,
      iv. g is superadditive.
   (e) Let $\mathcal{F}$ be the set of all real-valued functions h on $\mathbb{R}^m$ that satisfy $h(0) = 0$, h is superadditive and h is non-decreasing function. Let

$$\begin{aligned} \rho = \quad & \min \quad h(b) && (7.25) \\ & s.t. \quad h(A^j) = c_j \ \forall j \in \{1, \ldots, n\} && (7.26) \\ & \qquad h(-A^j) = -c_j \ \forall j \in \{1, \ldots, n\} && (7.27) \\ & \qquad h \in \mathcal{F}, && (7.28) \end{aligned}$$

[1] The function f is often called as value function of an IP.

where A^j is the j^{th} column of A and c_j is the j^{th} component of c. Prove that following duality results:

i. Weak Duality: $c^T x \leq h(b)$ for any $x \in S^b$ and any function h satisfying (7.26) - (7.28).

ii. Strong Duality: If $b \in S$ and $f(b)$ is finite, then $\rho = f(b)$.

Chapter 8

Branch and Bound, Presolve, Primal Heuristics

8.1 Branch and Bound Algorithm

Please refer to the textbook for the detail description of Branch and Bound Algorithm. Here we discuss some flexibilities that we can control in B&B.

8.1.1 Variable Selection

Suppose we have chosen an active node i. Associated with it is the linear programming solution x^i. Next we must choose a variable to define the division. We restrict it to the index set $N^i = \{j \in N \mid x^i_j \notin \mathbb{Z}\}$. Empirical evidence shows that the choice of a $j \in N^i$ can be very important to the running time of the algorithm. Among different variable selection rules, we discuss the following.

1. **Strong Branching.** Strong branching chooses as the branching variable the variable such that it maximizes the estimated improvement in the objective value. It first generates a list of candidates, then branches on each candidate and records the improvement in the objective value. The candidate with the largest improvement is chosen as the branching variable. Strong branching can be effective for combinatorial problems, but it is usually computationally expensive.

2. **Pseudo-Cost Approach.** This method chooses as the branching variable the variable such that it maximizes the weighted up and down pseudocosts. Pseudocost branching attempts to branch on significant variables first, quickly improving lower bounds. Pseudocost branching estimates significance based on historical information; however, this approach might not be accurate for future search.

3. **Priority.** If we have specific knowledge of the problem, we can use it to determine which variable to branch on. We can branch on the important variables first, e.g., first decide which warehouses to open, then decide the vehicle routing; branch on earlier (time-based) decisions first, etc. There are mechanisms for giving the variables a priority order, so that if two variables are fractional, the one with the high priority is branched on first.

4. **Generalized Upper Bound (GUB) Constraints.** Many integer programs with binary variables have generalized upper bound constraints of the form

$$\sum_{j \in J} x_j = 1,$$

where J is some index subset with very large cardinality. Instead dividing the feasible region based on the value of a specific variable, which may lead to an unbalanced tree, it is commonly more desirable to divide the feasible region of the parent roughly equally between the children. In particular, we consider the following branching rule:

$$\sum_{j \in J_1} x_j = 0 \quad \text{or} \quad \sum_{j \in J_2} x_j = 0,$$

where J_1, J_2 is a partition of J. Note that it seems reasonable to have J_1 and J_2 of nearly equal cardinalities.

8.1.2 Node Selection

Given a list of active subproblems, or equivalently, a partial tree of active nodes, the question is to decide which node should be examined in detail next. Here are two basic options.

1. **Best Upper Bound.** This selection rule chooses the node with the smallest (or largest, in the case of a maximization problem) relaxed objective value. The best upper bound strategy tends to reduce the number of nodes to be processed and can improve lower bounds quickly. However, this does not necessarily find feasible solutions quickly and can result in the solver running out of memory.

2. **Depth-First Search.** This rule chooses the node that is deepest in the search tree. Depth-first search is effective in locating feasible solutions, since such solutions are usually deep in the search tree. Compared to the costs of the best upper bound strategy, the cost of solving LP relaxations is less in the depth-first strategy. The number of active nodes is generally small, but it is possible that the depth-first search will remain in a portion of the search tree with no good integer solutions. This occurrence is computationally expensive.

8.2 Preprocessing and Probing

Preprocessing of an IP model consists of the tecniques such as

- Detecting redundant constraints
- Variable fixing
- Checking infeasibility

Let $Ax + Gy \leq b, x \in \{0,1\}^n, y \in \mathbb{R}^p_+$
Take the l'th row:

$$\sum_{j \in B^+} a^l_j x_j - \sum_{j \in B^-} a^l_j x_j + \sum_{j \in C^+} g^l_j y_j - \sum_{j \in C^-} g^l_j y_j \leq b^l$$

Define P^l as

- All constraints except l'th row
- All bounds
- All integrality constraints, non-negativity constraints

Feasibility check

Now consider the following problem:

$$min \left(\sum_{j \in B^+} a_j^l x_j - \sum_{j \in B^-} a_j^l x_j + \sum_{j \in C^+} g_j^l y_j - \sum_{j \in C^-} g_j^l y_j \right) \tag{8.1}$$
$$s.t. (x, y) \in P^l \tag{8.2}$$

If $z^* > b^l \Rightarrow$ infeasible

Suppose $z^{LP} \leq z^*$ and $z^{LP} > b^l = 1$ infeasible.

Evaluate

$$\sum_{j \in B^+} a_j^l 0 - \sum_{j \in B^-} a_j^l 1 + \sum_{j \in C^+} g_j^l l_j - \sum_{j \in C^-} g_j^l u_j > b^l$$

This detects infeasibility very quickly!

Detecting redundant constraints

Let

$$max \left(\sum_{j \in B^+} a_j^l x_j - \sum_{j \in B^-} a_j^l x_j + \sum_{j \in C^+} g_j^l y_j - \sum_{j \in C^-} g_j^l y_j \right) \tag{8.3}$$
$$s.t. (x, y) \in P^l \tag{8.4}$$

If $z^* \leq b^l \Rightarrow$ inequality redundant

$z^{UB} \geq z^*$ and $z^{UB} \leq b^l \Rightarrow$ inequality redundant

Evaluate

$$\sum_{j \in B^+} a_j^l + \sum_{j \in C^+} g_j^l u_j - \sum_{j \in C^-} g_j^l l_j \leq b^l$$

Variable bound improvement

Let

$$z^* = max\left(\sum_{j \in B^+} a_j^l x_j - \sum_{j \in B^-} a_j^l x_j + \sum_{j \in C^+/\{k\}} g_j^l y_j - \sum_{j \in C^-} g_j^l y_j\right) \quad (8.5)$$

$$s.t.(x,y) \in P^l \quad (8.6)$$

Then

$-z^* + g_k^l y_k^l \leq b^l$

$y_k \leq \dfrac{b^l - z^*}{g_k^l} < u_k$. If this holds, it can improve the upper bounds.

Variable Fixing

Similarly,

$$z^* = max\left(\sum_{j \in B^+} a_j^l x_j - \sum_{j \in B^-} a_j^l x_j + \sum_{j \in C^+} g_j^l y_j - \sum_{j \in C^-} g_j^l y_j\right) \quad (8.7)$$

$$s.t.(x,y) \in P^l \quad (8.8)$$

$$x_k = 0 \quad (8.9)$$

$z^* > b^l$

If $z_{lb} \leq z^*$ and $z^* > b^l \Rightarrow$ fix $x_k = 1$

Probing

$$z^* = max\left(\sum_{j\in B^+} a_j^l x_j - \sum_{j\in B^-} a_j^l x_j + \sum_{j\in C^+} g_j^l y_j - \sum_{j\in C^-} g_j^l y_j\right) \tag{8.10}$$

$$s.t.(x,y)\in P^l \tag{8.11}$$

$$x_{k_1} = 1, x_{k_2} = 1 \tag{8.12}$$

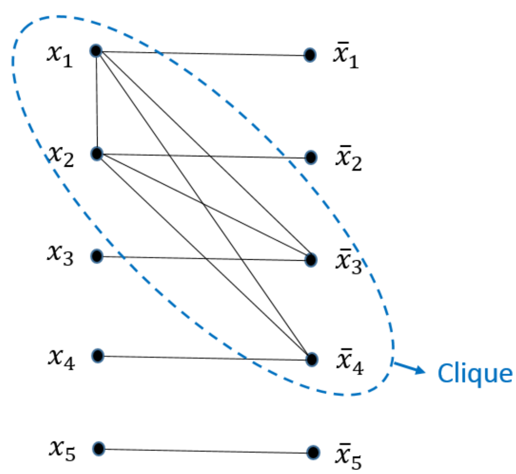
$$z_{lb} \leq z^*$$

$$z_{lb} > b^l$$

Hence, $x_{k_1} + x_{k_2} \leq 1$

Conflict Graph

Search for cliques in the conflict graph.



$x_1 + x_2 + \bar{x_3} + \bar{x_4} \leq 1$ This is a strong inequality.

$x_1 + x_2 - x_3 - x_4 \leq -1$

These inequalities are called **clique inequalities**.

$$\sum_{i \in L} x_i + \sum_{j \in R} \bar{x}_i \leq 1$$

If L & R have exactly one variable in common, then we can fix the variables in L (except common variable) to 0, and fix the variables (except common variable) in R to 1.

If L and R have more than one variable in common, then the problem is infeasible.

8.3 Primal Heuristics

Primal Heuristics are methods that look for feasible solutions, with no guarantees of any kind. In principle, the running time of the heuristics must be controlled. Primal heuristics are needed to : Prove feasibility of the model, Speed up search, and for Primal bound needed for pruning for branch and bound tree. Primal heuristics are quite good in commercial solvers, often finding optimal solutions (or close to optimal solutions) very quickly.

Primal heuristics can be broadly classified intwo two categories:

-
-

Bibliography