Integer Programming (IP) is the natural way of modeling many real-world and theoretical problems, including some combinatorial optimization problems, and it is a broad and well-studied area with a lot of potential to improve. In this class, we plan to study the following topics, which would provide the basic fundamental knowledge for further studies in IP:

- **Introduction**: IP techniques and complexity
- **Well-solved problems**: Which problems are “nice”?
- **Polyhedral theory**: Theoretical background for defining polyhedrons by facets or by extreme points and rays
- **Valid inequalities**: The main theoretical block necessary for cutting plane algorithms.
  - Basic knowledge such as separation and lifting
  - Problem-specific cuts (such as cover cuts)
  - General IP cuts: Gomory, Lift&Project cuts
- **Decomposition methods**: Dealing with large problems in smart ways
  - Column generation and Branch&Price
  - Benders’ Decomposition
- **Relaxations and duality**: Fundamental knowledge on bounds
  - Lagrangian relaxation and duality
  - LP and combinatorial relaxations, surrogate dual
- **IP heuristics**: Practical methods with no guarantees

**Prerequisites**: 620-362 recommended - basic knowledge of linear algebra, linear programming (especially the simplex method), and network models will be expected throughout the course. For an overview of these subjects, you can refer to Ch.2-4 and Ch. 7-8 of Winston [1994]. Knowledge of LP duality and dual-simplex method will also be helpful. (Refer to sections 6.5-6.11 of the same book if you are not sure)

**Textbooks**: Wolsey [1998] and Nemhauser and Wolsey [1988]

**Note**: These notes will not necessarily cover all material from the class but should be treated as supplementary. Class attendance is important for following the course.
1 Introduction

A Mixed Integer Programming (MIP) (“mixed” due to the fact that some of the variables are restricted to take only integer values) problem is an optimization problem with a linear objective function and linear constraints. (Note that we do not refer to problems with a nonlinear objective function or nonlinear constraints as MIP in this class - these problems are referred as MINLP in the literature.)

A typical MIP problem can be stated in the following form:

\[
\min_{x,y} \{ c^T x + h^T y | (x, y) \in X \}
\]

where \( X = \{ Ax + Dy \leq b, x \in \mathbb{R}^n, y \in \mathbb{Z}^p \} \)

The vectors \( x = (x_1, x_2, ..., x_n) \) and \( y = (y_1, y_2, ..., y_p) \) represent continuous and integer variables, respectively. Note that \( A \in \mathbb{R}^{m \times n} \) and \( D \in \mathbb{R}^{m \times p} \). Throughout the class, we will assume for simplicity all data to be integer.

If \( n = 0 \), then it is called Pure Integer Program. If \( y \in \{0, 1\}^p \), then it is called Binary Program. \( X \) represents the feasible region associated with this problem, and we will study it in detail in this class.

**Example:** Consider the feasible region defined by

\[
\begin{align*}
y_1 - 2y_2 & \geq -5 \\
2y_1 + 2y_2 & \leq 9 \\
6y_1 + 2y_2 & \leq 21 \\
y & \in \mathbb{Z}_+^2
\end{align*}
\]

Different than LP problems, the feasible region here is not a convex set but a set of discrete “regions”, in this case points.
When is integer programming necessary?

- “Linear” approximation is not good enough for modeling an entity (e.g., number of airplanes - fractional value is not meaningful!)
- “Setup decisions” have a binary character. (e.g., build a warehouse or not)

**Example:** *Knapsack problem*: Given a budget of $b$ and $n$ projects to choose from, each project $i$ with a price $a_i$ to invest in and an expected return $p_i$, how can we achieve the best investment strategy?

*Variables:* For each project $i \in [1, n]$, whether or not choose the project $i \Rightarrow y \in \{0, 1\}^n$ (alternatively, $y \in \mathbb{Z}_+^n$)

*Objective:* Maximize total return $\Rightarrow \max p^T y = \sum_{i=1}^n p_i y_i$

*Constraints:* Budget limitation $\Rightarrow a^T y = \sum_{i=1}^n a_i y_i \leq b$

### 1.1 Integer Programming Methods

For LP’s, we used Simplex method, thanks to some “nice” properties of the feasible region such as convexity and the fact that if there exist an optimal solution, it should be a corner of the polyhedron. Although MIP’s do not have these properties, some characteristics from linearity can still be used for IP solution methods.

**Example (cont’d):** Recall $X = \{y_1 - 2y_2 \geq -5, 2y_1 + 2y_2 \leq 9, 6y_1 + 2y_2 \leq 21, y \in \mathbb{Z}_+^2\}$. How does it compare to $\bar{X} = \{y_1 - y_2 \geq -2, y_1 + y_2 \leq 4, y_1 \leq 3, y \in \mathbb{Z}_+^2\}$?

What is the advantage of $\bar{X}$?
IP Methods can be classified as follows:

1. **Exact methods:** Branch&Bound, Cutting planes, Branch&Cut, Decomposition methods (Column generation, Benders',...)

2. **Non-exact methods:** IP heuristics, Metaheuristics (Tabu search, simulated annealing, genetic algorithms, ...), problem-specific heuristics

Differences between these classes: Solution guarantee and quality, solution times and practicality, ease of implementation, ...

1.1.1 **Branch&Bound**

This is the most traditional IP method dividing the problem into smaller and easier sub-problems, with the aim of “smart enumeration” instead of “complete enumeration” (why?). It works as follows: Solve the LP relaxation of MIP. If the solution is fractional, then **branch** on a integer variable having a fractional value. Solve these subproblems and use their results for **bounding** purposes. Record this process as a tree, where “child nodes” are simply subproblems. **Prune** nodes by optimality, infeasibility or bound.

**Example (cont’d):** Graphical view of B&B on the previous feasible set, with the given dashed objective function to maximize.

How can we draw the B&B tree?

**Example:** Section 7.3 of Wolsey [1998]
1.1.2 Cutting Plane Methods and Branch&Cut

We can add cuts (i.e., valid inequalities) to the formulation to make it “stronger”. There are problem-specific and general purpose cuts for MIP’s (more detail later in the course). We can see a cut as any inequality that cuts off a part of the LP relaxation region that does not contain any integer point - see the figure below. In theory, we can add all cuts so that we obtain an “integral” polyhedron (What about practicality? Do we need to add all cuts?).

Both pure B&B and pure cutting plane approaches are too naive to be efficient in solving general MIP problems. Combining these two methods results in Branch&Cut, where cuts are added to each node of the B&B tree to obtain better bounds and hence more efficient pruning.

1.2 Brief Overview of Complexity

This is intended to give a quick and shallow overview of complexity to understand better why MIP’s are hard to solve (complexity is itself a big research area!). Complexity theory provides the base of studying integer programming.

Recall the standard MIP: \[ \min_{x,y} \{ c^T x + h^T y | Ax + Dy \leq b, x \in \mathbb{R}^n, y \in \mathbb{Z}^p \} \]. What is the “length” \( L_{MIP} \) of the input of this MIP, i.e., how much space it takes on a computer?

\[
L_{MIP} = \sum_{i=1}^{n} \log_2 |c_i| + \sum_{i=1}^{p} \log_2 |h_i| + \sum_{i=1}^{m} \sum_{j=1}^{n} \log_2 |A_{ij}| + \sum_{i=1}^{m} \sum_{j=1}^{p} \log_2 |D_{ij}| + \sum_{i=1}^{m} \log_2 |b_i|
\]

**Definition 1** The function \( f(n) \) is said to be \( O(g(n)) \) if \( \exists C, \varepsilon > 0 \) such that \( |f(n)| \leq C|g(n)| \) for any \( n > \varepsilon \).
Example: $5n^2$ is $O(n^2)$, $-7n^3$ is not $O(n^2)$, $-2^n$ is $O(2^n)$, $0.5n$ is not $O(\log(n))$.

The big-O notation is crucial for understanding time complexity of an algorithm.

Definition 2 Let $f(L)$ be the worst-case number of basic operations of an algorithm $A$ on an input of length $L$. If $f(L)$ is $O(L^k)$ for some $k > 0$, then $A$ is said to be a polynomial-time algorithm. If $f(L)$ is $O(k^L)$ for some $k > 1$, then $A$ is said to be an exponential-time algorithm.

Even though in practice we might observe a polynomial-time algorithm and an exponential-time algorithm taking similar times for many problem instances, it is important to note that the growth is very steep for exponential problems, e.g., $20^2 = 400$ whereas $2^{20} > 1$ million.

Definition 3 If $f(L)$ is $O(\ell^k)$ for some $k > 0$, where $\ell$ is the unary value of some data of the problem, then $A$ is said to be a pseudopolynomial-time algorithm.

To better understand what $\ell$ represents, recall the definition of $L_{MIP}$. Problems with pseudopolynomial algorithms are one of the main motivations for studying integer programming.

Example: Integer knapsack problem and pseudopolynomial algorithm

We will conclude this section with the following formal definitions.

Definition 4 Let $P$ an optimization problem with over a feasible set $X$. $P$ is in class $\mathcal{NP}$ if there is a polynomial-time algorithm to check whether $x \in X$. $P$ is in class $\mathcal{P}$ if there is a polynomial-time algorithm that solves $P$.

If a problem class $P$ is not known to have a polynomial algorithm to solve, then it is referred as $\mathcal{NP}$-hard. Note that Simplex method has exponential-time complexity in worst-case, although it is very efficient in practice and is polynomial in general. LP’s can be solved in polynomial time (ellipsoid algorithms), and most MIP’s are $\mathcal{NP}$-hard.
2 Well-Solved Problems

We talked before about the importance of formulations, and emphasized that if the LP relaxation of an MIP has all integer extreme points, i.e., if it is an integral polyhedron, we can simply apply LP techniques to solve these problems. We now investigate some important structures and learn which problems have integral polyhedron property naturally. Review Ch.3 of Wolsey [1998] for details; it’s brief and compact. Also recall that we assume all data to be integral.

**Definition 5** A matrix is totally unimodular (TU) if all its square submatrices have determinant equal to -1,0 or 1.

**Proposition 6** The following statements are equivalent:

1. $A$ is TU.
2. $A^T$ is TU.
3. $(A, I)$ is TU.

Next, we present the key result why these TU matrices are important for our purposes.

**Proposition 7** Let $A \in \{-1, 0, 1\}^{m \times n}$ be TU and $b \in \mathbb{Z}^m$. Then, $X = \{x \in \mathbb{R}^n | Ax \leq b, \}$ is an integral polyhedron if $X \neq \emptyset$.

*Proof.* From the Simplex method, we know that for each vertex $x = \{x_B, x_N\}$, $[A, -I] = [B, N]$ and $Bx_B + Nx_N = b$ hold, where $B$ is a basis matrix. Hence, $x_B = B^{-1}b$. The integrality of $B^{-1}$ follows Cramer’s rule. □

How can we easily check whether a matrix is TU or not? The following provides a simple sufficient condition.

**Proposition 8** A matrix $A \in \{-1, 0, 1\}^{m \times n}$ is TU if each column has at most two nonzero entries and there is a partition of rows such that for each column with two nonzero entries, these entries are in the same partition if they have different signs and they are in different partitions otherwise.

Where can we use this property for recognizing TU matrices?

**Example:** Minimum cost network flow problems - can we prove integrality? Where else can we generalize this?
3 Polyhedral Theory

Here, we review some basics related to polyhedra. Although the feasible set of an MIP is not a convex set, this section is crucial to understand the foundations of the IP theory since it is based on defining this set as a continuous set defined by linear inequalities. We start with basic definitions.

**Definition 9** Let \( x_1, x_2, \ldots, x_k \in \mathbb{R}^n \) be given points. If the equation \( \sum_{i=1}^{k} \lambda_i x_i = 0 \) has the unique solution of \( \lambda = 0^k \), then these \( k \) points are **linearly independent**. If the system \( \{ \sum_{i=1}^{k} \lambda_i x_i = 0; \sum_{i=1}^{k} \lambda_i = 0 \} \) has the unique solution of \( \lambda = 0^k \), then these \( k \) points are **affinely independent**.

**Proposition 10** The following statements are equivalent:

1. \( x_1, x_2, \ldots, x_k \in \mathbb{R}^n \) are affinely independent.
2. \( 0, x_2 - x_1, \ldots, x_k - x_1 \) are affinely independent.
3. \( x_2 - x_1, \ldots, x_k - x_1 \) are linearly independent.
4. \( [x_1, -1], [x_2, -1], \ldots, [x_k, -1] \) are linearly independent.

Proposition 10 is useful when one is trying to find affinely independent points but they are not as obvious as linearly independent ones. Also note that \( k \leq n + 1 \).

**Definition 11** Let \( x_1, x_2, \ldots, x_k \in \mathbb{R}^n \) be points. \( x \) is a **convex combination** of these points if there exist nonnegative numbers \( \lambda_i \) such that \( \sum_{i=1}^{k} \lambda_i x_i = x \) and \( \sum_{i=1}^{k} \lambda_i = 1 \).

**Definition 12** Let \( C \subseteq \mathbb{R}^n \) be a set. \( C \) is a **convex set** if any convex combination of any two points \( x_1, x_2 \in C \) is in the set \( C \).

**Definition 13** \( C \) is a **polyhedron** if \( C \subseteq \mathbb{R}^n \) is a set of points that satisfies a finite number of linear inequalities. A bounded polyhedron is called a **polytope**.

**Proposition 14** A polyhedron is a convex set.

This simple fact is a key in the establishment of polyhedral theory and it ensures solution procedures of Linear Programming (LP) to find the optimal solution. (We do not review them here, but convexity of a set provides a crucial base for global optimal solutions in NLP.) The following is crucial for relating the feasible set of an MIP problem to the polyhedral theory.

**Definition 15** Given a set \( C \subseteq \mathbb{R}^n \), the **convex hull** of \( C \), denoted as \( \text{conv}(C) \), is the set of all points that are convex combinations of points in \( C \).
Let \( X \) be the feasible region of the general MIP problem. The following are important results why we apply polyhedral theory to MIP’s.

**Proposition 16** The convex hull of \( X \) is a polyhedron.

**Proposition 17** Solving an MIP problem is equivalent to solving it over the convex hull of its feasible region, i.e.,

\[
\min_{x,y} \{ c^T x + h^T y \mid (x, y) \in X \} = \min_{x,y} \{ c^T x + h^T y \mid (x, y) \in \text{conv}(X) \}
\]

Proposition 17 is the key result for MIP’s and it will be extended for different use in the next sections. Note that \( \text{conv}(X) \) seems to be the “ideal formulation” of the MIP problem, so one could insist on defining \( \text{conv}(X) \) to solve an MIP problem. However, this has two potential drawbacks: 1) It is generally very difficult to generate all or even some of the inequalities that define \( \text{conv}(X) \). 2) \( \text{conv}(X) \) may be defined by an exponential number of inequalities.

Next, we discuss dimension and its importance.

**Definition 18** The **rank** of a matrix \( A \), denoted as \( \text{rank}(A) \), is the maximum number of linearly independent columns (= rows) of it.

**Proposition 19** If \( C = \{ x \in \mathbb{R}^n \mid Ax = b \} \neq \emptyset \), then the maximum number of affinely independent points in \( C \) are \( n + 1 - \text{rank}(A) \).

**Definition 20** The **dimension** of a polyhedron \( C = \{ x \in \mathbb{R}^n \mid Ax \leq b \} \), denoted as \( \text{dim}(C) \), is one less than the maximum number of affinely independent points in \( C \). If \( \text{dim}(C) = n \), then \( C \) is called to be **full-dimensional**.

Why not use linearly independent points instead of affinely independent points for defining dimension? Discuss...

**Example:** Feasible set of the node-packing problem is defined as \( X = \{ y \in \{0,1\}^{\mid V\mid} \mid y_i + y_j \leq 1 \ \forall (i,j) \in E \} \), for a given graph \( G = (V,E) \). Consider the claim \( \text{dim}(\text{conv}(X)) = \mid V\mid \). What do we need to find? □

**Example:** Recall the 0-1 knapsack problem set, \( X = \{ y \in \{0,1\}^n \mid \sum_{i=1}^{n} a_i y_i \leq b \} \).
Claim: \( \text{conv}(X) \) is full-dimensional if and only if \( a_i \leq b, \ \forall i \in \{1,...,n\} \). □

In general, most MIP convex hulls are not full-dimensional, e.g., we needed an extra requirement even in 0-1 knapsack problem. This relates to the fact that at least an inequality of the system of inequalities is satisfied as an equation for all the points in the polyhedron. Let \( C = \{ x \in \mathbb{R}^n \mid Ax \leq b \} \) for the rest of this subsection. Consider the partitioning of all rows of \( (A,b) \) to \( (A^=, b^=) \) and \( (A^\leq, b^\leq) \), where \( (A^=, b^=) \) includes all rows that are satisfied as equation for all \( x \in X \).
Definition 21 \( \bar{x} \in C \) is an **inner point** of \( C \) if \( A \leq \bar{x} < b \). \( \bar{x} \in C \) is an **interior point** of \( C \) if \( A \bar{x} < b \).

Proposition 22 If \( C \neq \emptyset \), then \( C \) has an inner point. \( C \) is full-dimensional if and only if \( C \) has an interior point.

The last result is derived from the following proposition, which is crucial for determining the dimension of polyhedrons that are not full-dimensional.

Proposition 23 \( \dim(C) + \text{rank}(A^n, b^n) = n \)

*Proof.* Using Proposition 19...

**Example:** Example 2.1 of Nemhauser and Wolsey [1988]

**Example:** Uncapacitated Facility Location (UFL) problem can be stated as follows:

There are \( M \) clients and \( N \) potential locations for depots. Clients can satisfy their demand \( d_i \) from any combination of depots, where per unit shipment cost from depot \( j \) \( (c_{ij}) \) and fixed opening cost of a depot \( j \) \( (h_j) \) are charged appropriately. Then, using binary variables \( y_j \) to indicate whether the depot \( j \) is opened or not and continuous variables \( x_{ij} \) indicating the fraction of \( d_i \) satisfied by depot \( j \), the problem can be formulated as follows:

\[
\min_{x,y} \sum_{i=1}^{M} \sum_{j=1}^{N} d_i c_{ij} x_{ij} + \sum_{j=1}^{N} h_j y_j \tag{1}
\]

s.t. \( \sum_{j=1}^{N} x_{ij} = 1 \quad \forall i \in \{1, \ldots, M\} \tag{2} \)

\[
\sum_{i=1}^{M} x_{ij} \leq M y_j \quad \forall j \in \{1, \ldots, N\} \tag{3}
\]

\( x \in \mathbb{R}^{M \times N}, y \in \{0, 1\}^{N} \tag{4} \)

Let \( X_{UFL} \) be the feasible set defined by (2)-(4). Prove that \( \dim(\text{conv}(X_{UFL})) = N + MN - M \).

First, note that since we already have \( M \) equations and since there are \( N + MN \) variables, by Proposition 23, \( \dim(\text{conv}(X_{UFL})) \leq N + MN - M \). Now, we need to show that \( \dim(\text{conv}(X_{UFL})) \geq N + MN - M \), i.e., find \( N + MN - M + 1 \) affinely independent points. First consider the case \( N = 2 \). What can you say about the following \( M + 3 \) points?

\[
y = (1, 0); x = (1, 1, \ldots, 1; 0, 0, \ldots, 0)
\]

\[
y = (0, 1); x = (0, 0, \ldots, 0; 1, 1, \ldots, 1)
\]

\[
y = (1, 1); x_i = (0, 1)(\forall i \in \{1, \ldots, \bar{i}\}); x_{\bar{i}} = (1, 0)(\forall i \in \{\bar{i} + 1, \ldots, M\}), \forall \bar{i} \in \{0, \ldots, M\}
\]

What about \( N > 2 \)?

We are interested in the two ways to describe a polyhedron: By facets or by extreme points and rays. Next, we discuss these in detail.
3.1 Defining Polyhedra by Facets

**Definition 24** The inequality $\pi^T x \leq \pi_0$ is a **valid inequality** for $C = \{ x \in \mathbb{R}^n | Ax \leq b \}$ if and only if it is satisfied by all points of $C$.

An important remark for MIP’s is stated as follows:

**Corollary 25** If an equality is valid for a feasible set $X$, then it is also valid for $\text{conv}(X)$.

It is not clear yet how we can generate “valid inequalities” (coming soon!). Even though we mentioned before that describing the convex hull of an MIP is in general hard, it is computationally useful to generate at least some good valid inequalities to improve the quality of the formulation. But how can we define good?

**Definition 26** Let $\alpha^T x \leq \alpha_0$ and $\beta^T x \leq \beta_0$ be valid inequalities for $C$. The former inequality **dominates** the latter if $\exists \mu > 0$ such that $\alpha \geq \mu \beta$ and $\alpha_0 \leq \mu \beta_0$.

It is crucial to compare different inequalities to eliminate redundant constraints and hence improve the formulation. Next, we will discuss what are the best valid inequalities we can use.

**Example:** Recall UFL. Consider the inequality $x_{ij} \leq y_j$, $\forall i \in \{1, ..., M\}, j \in \{1, ..., N\}$. What can you say about it?

**Example:** Recall node packing problem. Discuss briefly **odd-hole inequalities**.

**Definition 27** Let $\pi^T x \leq \pi_0$ be a valid inequality for $C$. Then, $F = \{ \bar{x} \in C | \pi^T \bar{x} = \pi_0 \}$ is a **face** of $X$. Moreover, $F$ is called to be **proper** if $F \neq \emptyset$ and $F \neq C$. If $\dim(F) = \dim(C) - 1$, then $F$ is a **facet** of $C$.

**Proposition 28** Let $F$ be a proper face of $C$ but not a facet. Then, an inequality defining $F$ is not necessary for the description of $C$.

Facet-defining inequalities dominate any other valid inequalities and hence it is in our best interests to be able to generate (explicitly or implicitly) at least some of them for more efficient solutions. Also note that inequalities defining high-dimensional faces can be useful computationally as well, even though they are not necessary for the definition of the polyhedron.

**Example:** Recall the MIP problem $X = \{ y \in \mathbb{Z}_+^2 | y_1 - 2y_2 \geq -5, 2y_1 + 2y_2 \leq 9, 6y_1 + 2y_2 \leq 21 \}$ and consider $\text{conv}(X)$ (shaded area in the figure below).
Facets are indicated with brackets and zero-dimensional faces with hollow points. Let’s discuss inequalities defining faces and facets...

**Example:** Recall the inequalities $x_{ij} \leq y_j$, $\forall i \in \{1, \ldots, M\}, j \in \{1, \ldots, N\}$ for UFL. Are these necessary for describing $\text{conv}(X_{UFL})$?

### 3.2 Defining Polyhedra by Extreme Points and Rays

**Definition 29** Let $C = \{x \in \mathbb{R}^n | Ax \leq b\}$. A point $\bar{x} \in C$ is an **extreme point** of $C$ if $\bar{x}$ cannot be written as a convex combination of any other points in $C$.

Extreme points can also be seen as 0-dimensional faces of a polyhedron (see the figure above).

**Definition 30** If $C = \{x \in \mathbb{R}^n | Ax \leq b\} \neq \emptyset$, any extreme point (except origin) of the polyhedron $\{x \in \mathbb{R}^n | Ax \leq 0\}$ is an **extreme ray** of $C$.

A polyhedron has a finite number of extreme points and rays, which follows the fact that the number of faces is finite for any polyhedron. We will also review Minkowski’s theorem and Weyl’s theorem here (refer to pp.96-98 of Nemhauser and Wolsey [1988]) - these prove that a polyhedron can be written by its extreme points and rays and vice versa.
4 Valid Inequalities

We already discussed briefly why valid inequalities play an important role for solving MIP’s, and we also reviewed important terminology such as dominance and facet-defining inequalities and their importance. Now, we handle the issue of “valid inequalities” from the point of being able to define inequalities for a particular or general problem to being able to generate those inequalities. First, we need some basics.

Let $z$ be the optimal objective function value of the MIP, and let superscript $LP$ indicate LP relaxation, e.g., $z_{LP}$ refers to the objective function value of the LP relaxation. We know that $z \geq z_{LP}$ since $\text{conv}(X) \subseteq X^{LP}$, and therefore we want to cut off $X^{LP} \setminus \text{conv}(X)$, assuming the general case of $X^{LP} \neq \text{conv}(X)$.

**Definition 31** Given a point $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^p$, the separation problem for $X$ is as follows: Either (1) confirm $(x^*, y^*) \in \text{conv}(X)$ or (2) find a violated inequality for $(x^*, y^*)$, i.e., a valid inequality $\pi x + \mu y \leq \pi_0$ for $X$ such that $\pi x^* + \mu y^* > \pi_0$.

Although this definition is a formal one, it is crucial to understand the context of valid inequalities and cutting plane methods. Cutting plane methods solve the LP relaxation (or any other relaxation, for that matter) of the MIP problem, and then the solution obtained $(x^*, y^*)$ is checked whether $(x^*, y^*) \in \text{conv}(X)$ holds or not. In practice, this separation procedure is dependent on the family of valid inequalities we are trying to generate. In this approach, only the valid inequalities that are violated by the LP relaxation solution are generated and added to the formulation. Therefore, cutting plane methods do not necessarily provide the full description of the convex hull, but that is also not necessary to solve an optimization problem. Also note that in commercial solvers, cutting planes are usually employed within the Branch&Bound Algorithm, resulting in the “Branch&Cut” (B&C) algorithm, for more efficient solution processes.

A significant result from the complexity theory that relates the optimization and separation problems is presented next.

**Proposition 32** Optimization problem and separation problem are polynomially equivalent, i.e., the following statements are equivalent:

- Solving $\min_{x, y} \{c^T x + h^T y | (x, y) \in X\}$ is solvable in polynomial time.
- Separating $(x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^p$ over $\text{conv}(X)$ is solvable in polynomial time.

Next, we will make an important definition from polyhedral theory that relates “subspaces”.

**Definition 33** Let $C = \{x \in \mathbb{R}^n, y \in \mathbb{R}^p | Ax + Dy \leq b\}$. Then, the projection of $C$ onto $x$, denoted as $\text{proj}_x(C)$, can be defined as follows:

$$\text{proj}_x(C) = \{x \in \mathbb{R}^n | \exists y \in \mathbb{R}^p \text{ such that } Ax + Dy \leq b\}$$
This is crucial for many IP techniques, such as extended reformulations, and we will see applications of it in the future. Finally, we make the following definition, where an inequality is extended from a lower dimension to a higher one (if possible).

**Proposition 34** Let $S_{n-1} = \{ x \in \{0,1\}^{n-1} | Ax \leq b \}$ and $S_n = \{ x \in \{0,1\}^n, y \in \{0,1\} | Ax + dy \leq b \}$. Assume $\pi^T x \leq \pi_0$ is a valid inequality for $S_{n-1}$. Then, lifting this inequality up to $S_n$ can be described as follows: If $S_n \cap \{ y = 1 \} = \emptyset$, then $y \leq 0$ is a valid inequality for $S_n$; otherwise, $\pi^T x + \pi_n y \leq \pi_0$ is a valid inequality for $S_n$ for any $\pi_n \leq \pi_0 - \max \{ \pi^T x | Ax \leq b - d, x \in \{0,1\}^{n-1} \}$.

Note that there are valid inequalities for general MIP problems, as well as specific families of valid inequalities for some particular types of problems. We will look into both of these in this course, starting with problem-specific cuts first.

### 4.1 Problem-specific Inequalities

#### 4.1.1 0-1 Knapsack Problem

Recall the 0-1 knapsack problem defined over the set $X_K = \{ y \in \{0,1\} | \sum_{i=1}^{n} a_i y_i \leq b \}$, and we will refer to $\text{conv}(X_K)$ as 0-1 knapsack polytope from now on. Assume again that $a_i \leq b$ holds for all $i \in \{1,...,n\}$. The main motivation for studying this polytope is that it appears as a subproblem in almost all MIP problems. Recall that $\text{dim}(\text{conv}(X_K)) = n$.

**Exercise:** What can you say about the “trivial” facets of this polytope? \hfill \Box

Next, we look into more interesting properties of this polytope.

**Definition 35** A cover of $X_K$ is a set $C \subseteq \{1,...,n\}$ such that $\sum_{i \in C} a_i > b$. $C$ is called a minimal cover if no proper subset of it defines a cover.

**Proposition 36** Let $C$ be a cover of $X_K$. Then, $\sum_{i \in C} y_i \leq |C| - 1$ is a valid inequality for $X_K$.

We will refer to this family of inequalities as “cover inequalities”.

**Example:** Consider the example 9.6 of Wolsey [1998]:

$$X = \{ x \in \{0,1\}^7 | (11,6,6,5,5,4,1)x \leq 19 \}$$

Can we find any cover/minimal cover inequalities? \hfill \Box

**Definition 37** The extended cover of a given minimal cover $C$, denoted as $E(C)$, is the set such that $C \subseteq E(C)$ and all $j \in \{1,...,n\} \setminus C$ such that $a_j \geq \max_{i \in C} \{ a_i \}$ belong to $E(C)$.  

14
Proposition 38  Let $E(C)$ be an extended cover for $X_{KN}$. Then, $\sum_{i \in E(C)} y_i \leq |C| - 1$ is a valid inequality for $X_{KN}$.

Obviously, extended cover inequality dominates the cover inequality for a given cover $C$. But how strong are these inequalities in general? Can we have a more systematic and efficient way of generating stronger inequalities? These are the questions we will try to answer.

Example (cont’d): Consider the cover $C = \{3, 4, 5, 6\}$. What is $E(C)$?

Proof. We need to show $n$ affinely independent points that satisfy the extended cover inequality as an equation. Consider the following points:

- $y_i = 0, y_j = 1$; for all $i \in C, j \in C\backslash i$ ($|C|$ points).
- $y_j = 1$ if $j \in (C \backslash \{p, p+1\}) \cup \{i\}$ and $y_j = 0$ otherwise; for all $i \in E(C) \backslash C$ ($|E(C)\backslash C|$ points)
- $y_j = 1$ if $j \in (C \backslash \{p\}) \cup \{i\}$ and $y_j = 0$ otherwise; for all $i \in \{1, \ldots, n\} \backslash E(C)$ ($n - |E(C)|$ points)

Example (cont’d): What can you say about the cover $C = \{3, 4, 5, 6\}$? What about $C = \{1, 2, 3\}$?

There are some other sufficient conditions (see e.g. Nemhauser and Wolsey [1988]) for cover inequalities to become facet-defining inequalities. Although extending a minimal cover is an efficient way to strengthen the cover inequalities, this is a limited method, as noted in the following example, and hence we will study lifting of cover inequalities.

Example (cont’d): Recall the cover $C = \{3, 4, 5, 6\}$ and $E(C)$. Consider the following inequality:

$$2y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \leq 3$$

Is it valid? How can we obtain it?
**Lifting Cover Inequalities**

Let $C$ be a minimal cover of $X_{KN}$. We are trying to find an inequality of the form:

$$
\sum_{i \in \{1, \ldots, n\} \setminus C} \alpha_i y_i + \sum_{i \in C} y_i \leq |C| - 1
$$

(5)

with the “best possible” values for $\alpha_i$, $i \in \{1, \ldots, n\} \setminus C$. We will obtain (5) by a *sequence-dependent* lifting procedure, so let’s look at the process at each step. Let $X_{KN}^C = \{y \in X_{KN} | y_j = 0 \ \forall j \in \{1, \ldots, n\} \setminus C\}$.

Obviously, the cover inequality defines a facet of $\text{conv}(X_{KN}^C)$ (why?). Now, consider a $j' \in \{1, \ldots, n\} \setminus C$ and let $X_{KN}^{C \cup j'} = \{y \in X_{KN} | y_j = 0 \ \forall j \in \{1, \ldots, n\} \setminus \{C \cup j'\}\}$. How high can we set $\alpha_{j'}$ such that

$$
\alpha_{j'} y_{j'} + \sum_{i \in C} y_i \leq |C| - 1
$$

(6)

is a valid inequality for $X_{KN}^{C \cup j'}$? There are two cases:

- $y_{j'} = 0$. Obviously, we can set $\alpha_{j'}$ as high as we want and it still would be valid.

- $y_{j'} = 1$. We need to make sure that for all points in $\{y \in X_{KN}^{C \cup j'} | y_{j'} = 1\}$, the inequality $\sum_{i \in C} a_i y_i \leq b - \alpha_{j'}$ holds. (why?)

This leads us to solving the problem:

$$
z_{j'} = \max\{\sum_{i \in C} y_i | \sum_{i \in C} a_i y_i \leq b - \alpha_{j'}, y_i \in \{0, 1\} \ \forall i \in C\}
$$

Then, setting $\alpha_{j'} = |C| - 1 - z_{j'}$ will ensure that (6) is valid for $X_{KN}^{C \cup j'}$. At this point, we have lifted $j'$th variable, and we can apply the same logic for all $j \in \{1, \ldots, n\} \setminus \{C \cup j'\}$.

**Example (cont’d):** Recall the cover $C = \{3, 4, 5, 6\}$ and $E(C)$. Consider lifting remaining variables in the order of 1, 2 and 7...

The next crucial question is how strong these inequalities obtained through this lifting procedure are, and the next result addresses that issue.

**Proposition 40** Let $C$ be a minimal cover, and assume $\alpha_i$ for all $i \in \{1, \ldots, n\} \setminus C$ are calculated using the above approach in any sequence. Then, (5) is facet-defining for $\text{conv}(X_{KN})$.

Note that different lifting sequences might produce different inequalities. This depends on the fact that if we lift a variable earlier, its coefficient will be at least as high as if we lifted that variable later. Finally, note that this process requires solving $n - |C|$ 0-1 knapsack problems to generate one lifted inequality.
**Separation Problem for Cover Inequalities**

Suppose that the LP relaxation of the knapsack problem provides us a fractional solution \( \bar{y} \), and if possible, we would like to find a cover inequality that cuts off this fractional point. So, the crucial question is that whether \( \bar{y} \) satisfies all cover inequalities or not. Formally speaking, separation question is: is there any set \( S \subseteq \{1,...,n\} \) such that \( \sum_{i \in S} \bar{y}_i > |S| - 1 \) and \( \sum_{i \in S} a_i > b \)?

First, note that \( \sum_{i \in C} y_i \leq |C| - 1 \equiv \sum_{i \in C}(1 - y_i) \geq 1 \). Hence, the separation problem can be written as follows:

\[
 z_{KN}^{SEP} = \min \left\{ \sum_{i=1}^{n} (1 - \bar{y}_i)z_i \mid \sum_{i=1}^{n} a_i z_i > b, z \in \{0,1\}^n \right\}
\]

If \( z_{KN}^{SEP} \geq 1 \), then \( \bar{y} \) satisfies all cover inequalities. Otherwise, what is the cover cut we generate? How much is the “violation”?

4.1.2 Symmetric Traveling Salesman Problem

Given a graph \( G = (V, E) \), traveling salesman problem (TSP) aims to find the least-cost tour that visits all the vertices. Such a tour (or cycle) visiting all vertices is called a Hamiltonian tour. We assume the graph to be undirected, hence it is symmetric - once we obtain a tour, it does not matter which direction we proceed. Defining binary variables \( y_e \) for each \( e \in E \) and edge costs (or travel times) \( c_e \), symmetric TSP can be formulated as follows:

\[
\begin{align*}
\min & \quad \sum_{e \in E} c_e y_e \\
st. & \quad \sum_{e \in \delta(i)} y_e = 2 \quad \forall i \in V \\
& \quad \sum_{e \in E(S)} y_e \leq |S| - 1 \quad \forall S \subset V \\
& \quad y \in \{0,1\}^{|E|}
\end{align*}
\]

Here, \( \delta(i) \) refers to all edges connected to vertex \( i \), and \( E(S) \) denotes all the edges connecting the vertices of \( S \) to \( S \). Constraint (8) ensures that each vertex is visited exactly once, and (8) are the subtour elimination constraints. Even though this is a correct formulation, it has the obvious issue that there are exponentially many subtour elimination constraints. When we assume \( G \) is a complete graph, these inequalities are facet-defining if \( 2 \leq |S| \leq |\lfloor |V|/2 \rfloor| \). Hence, we would like to generate them using a separation algorithm. An exact separation method is using the fractional solution as weights on edges, and then solve the general minimum cut problem and check whether the min cut causes a subtour or not.
4.2 General IP Inequalities

4.2.1 Integer Rounding: Chvátal-Gomory Procedure and MIR (Mixed Integer Rounding)

Here, the simple idea is that if we have a sum of integer variables, that value will be integer and hence we can round fractional values, i.e., if \( x \leq b \) and \( b \) is fractional, then obviously we can replace this inequality with \( x \leq \lfloor b \rfloor \). Recall that we used this idea before (e.g., odd-hole inequalities for node packing problem). Now the more crucial question is how to systematically use this to generate valid inequalities.

**Example:** Recall the problem we discussed before:

\[
\begin{align*}
y_1 - 2y_2 &\geq -5 \\
2y_1 + 2y_2 &\leq 9 \\
6y_1 + 2y_2 &\leq 21 \\
y &\in \mathbb{Z}^2_+
\end{align*}
\]

Now, note that \( 2y_1 + 2y_2 \leq 9 \) can be written as \( y_1 + y_2 \leq 4.5 \) and hence we can replace it with \( y_1 + y_2 \leq 4 \). Next, consider \( y_1 - 2y_2 \geq -5 \), which we can rewrite first as \( -y_1 + 2y_2 \leq 5 \), and then multiply by \( 1/2 \) and obtain \( -1/2y_1 + y_2 \leq 5/2 \). Hence, \( -y_1 + y_2 \leq 5/2 \) is a valid inequality (why?), which we can round to \( -y_1 + y_2 \leq 2 \).

Dashed lines in the figure indicate the inequalities we obtained by this rounding procedure. \( \square \)
The last procedure where we did not only round the right hand side of the inequality but also the coefficients of the variables is called Chvátal-Gomory procedure and can be generalized as follows for the pure integer set $X = \{ y \in \mathbb{Z}_+^n | Ay \leq b \}$, where $A = [a_1, a_2, ..., a_n]$ is an $m \times n$ matrix and $u \in \mathbb{R}_+^m$:

1. The inequality $\sum_{i=1}^n u^T a_i y_i \leq u^T b$ is valid since $u \geq 0$.
2. The inequality $\sum_{i=1}^n \lfloor u^T a_i \rfloor y_i \leq u^T b$ is valid since $y \geq 0$.
3. The inequality $\sum_{i=1}^n \lfloor u^T a_i \rfloor y_i \leq \lfloor u^T b \rfloor$ is valid since $y$ is integer.

This simple procedure is capable of generating all valid inequalities for an integer set in a finite number of steps. What prevents us from doing so?

**Example:** Example 8.10 of Wolsey [1998]

Next, we consider extending this simple logic to more general problems, i.e., MIP’s.

**Proposition 41** Let $X = \{ x \in \mathbb{R}_+, y \in \mathbb{Z} | y \leq b + x \}$, and assume $f = b - \lfloor b \rfloor > 0$. Then, the inequality

$$y \leq \lfloor b \rfloor + \frac{x}{1 - f}$$

is valid for $X$.

**Example:** Consider the set $\{ x \in \mathbb{R}_+, y \in \mathbb{Z} | 2y \leq 7 + 2x \}$. We can rewrite the only constraint as $y \leq 3.5 + x$ and hence we know that $y \leq 3 + 2x$.

Next, we will look into a more general mixed sets.

**Proposition 42** Let $X = \{ x \in \mathbb{R}_+, y \in \mathbb{Z}_+^2 | a_1 y_1 + a_2 y_2 \leq b + x \}$, and assume $f = b - \lfloor b \rfloor > 0$. Also let $f_i = a_i - \lfloor a_i \rfloor$ ($i = 1, 2$) and assume $f_1 \leq f \leq f_2$. Then, the MIR inequality

$$\lfloor a_1 \rfloor y_1 + (\lfloor a_2 \rfloor + \frac{f_2 - f}{1 - f}) y_2 \leq \lfloor b \rfloor + \frac{x}{1 - f}$$

is valid for $X$.

Note that these inequalities can be generalized simply to mixed sets with more integer and continuous elements.

**Example:** Consider the continuous 0-1 knapsack set $\{ s \in \mathbb{R}_+, y \in \{0, 1\}^3 | 2y_1 + y_2 + 2y_3 \leq \frac{7}{3} + s \}$. Consider Proposition 41. What inequality can we obtain? What if the coefficient of $y_3$ was $\frac{3}{2}$?

What can we say about the strength of these inequalities? Let’s briefly discuss Proposition 6.2 of Nemhauser and Wolsey [1988].
5 MIP Heuristics: A Brief Overview

So far in this lecture, we have seen many techniques to solve MIP problems, from Branch-and-bound to cutting planes and Branch-and-cut, from Lagrangian duality to decomposition techniques such as Benders. It is important to note that all these techniques we discussed are exact methodologies, i.e., they aim to find the optimal solution, no matter what it takes. The main advantage of these methodologies is that even if we terminate them before obtaining the optimal solution, we have information on solution quality. The obvious and significant disadvantage is that in practice, these techniques require generally significant enumeration (i.e., memory issues) and time (realistic problems usually never end up with the optimal solution).

This is the main motivation for heuristic methods, which can be described as “practical and intuitive” solution strategies. Heuristics are particularly useful when it comes to large scale problems from real world applications, when exact methods suffer most from enumeration and time issues. We can classify heuristics usually as “problem-specific” and “general MIP” heuristics. Specific problem structure plays an important role for the heuristics’ performance and hence there are many problem-specific methods in the literature. Note that combinations of different heuristics are more likely to provide better results. Even though fast solutions seem to be the main advantage of heuristics, it should be noted that the significant disadvantage of heuristics is the lack of knowledge about solution quality; there is even no guarantee on finding a feasible solution. However, note that some of the MIP heuristics we will examine here provide a bound, which is important for proving solution quality.

These tradeoffs between solution quality and solution time suggest us to combine different methods for efficient solution strategies. Therefore, the state of the art solvers in use today employ both exact methods and heuristics in sophisticated ways.

We will classify heuristics as follows in our context:

- **Construction heuristics**: These try to generate a solution from scratch.
- **Improvement heuristics**: Using known solution(s) for the problem, these try to obtain a better solution.

Before getting into more detail, let our MIP problem to be a binary mixed problem in the standard form:

\[
\min_{x,y} \{ c^T x + h^T y \mid (x, y) \in X \} \quad \text{where} \quad X = \{ Ax + Dy \leq b, x \in \mathbb{R}^n, y \in \{0, 1\}^p \}
\]

Note that this can be easily generalized to general integer variables (we consider here only binaries for simplicity).
5.1 Construction Heuristics

We will here review some basic heuristics such as rounding and diving, as well as more complicated ones like feasibility pump.

1. Rounding

- Let \((\bar{x}, \bar{y})\) be the solution of the LP relaxation.
- Set \(y_i = 0\) if \(\bar{y}_i \leq k\) and \(y_i = 1\) if \(\bar{y}_i \geq \ell\), for “some” \(0 \leq k \leq \ell \leq 1\)
- Then solve the LPR with these fixings again (reiterate if needed)

In general, rounding is too naive of an approach and hence often results in infeasibilities. It has been shown useful only for few specific problems, e.g. 2-OPT algorithm for vertex cover. Also, randomized/iterative rounding is widely used in approximation algorithms.

**Main use:** To attain quick solutions at the deep nodes of B&B

2. Diving

- Let \((\bar{x}, \bar{y})\) be the LP relaxation solution
- Let \(\bar{y}_i = \min_{i \in \{1, p\}} \min(\bar{y}_i, 1 - \bar{y}_i)\)
- Solve the LP:
  \[
  \min_{x,y} \{c^T x + h^T y | Ax + Dy \leq b, x \in \mathbb{R}^n, 0 \leq y \leq 1, y_i = |\bar{y}_i|\}
  \]
- If solution is not integral, go to step 1. Otherwise, STOP.

Diving is another basic idea that fixes one variable at a time (at each iteration) except in case of obtaining a fully integral solution in an intermediate step. It is a fast convergent method, although it has quite significant possibility of infeasibility as in rounding.

**Main use:** Depth-first search in B&B

3. LP-and-Fix

- Let \((\bar{x}, \bar{y})\) be the LPR solution
- Let \(Q = \{i | \bar{y}_i = 0\ or \ \bar{y}_i = 1\}\)
- Solve the MIP problem:
  \[
  \min_{x,y} \{c^T x + h^T y | Ax + Dy \leq b, x \in \mathbb{R}^n, y \in \{0, 1\}^p, y_i = \bar{y}_i \ \forall i \in Q\}
  \]
A very simple quick-and-dirty method related to diving, LP-and-fix can be found in almost all the commercial/academic solvers. The stronger the initial formulation, the more efficient LP-and-fix works.

4. **Relax-and-Fix**
   - Partition all binary variables into $k$ sets, i.e. $\bigcup_{i=1}^{k} Q_i = \{1, ..., p\}$
   - Let decisions in set $Q_i$ be more “important” than decisions in $Q_j$, for all pairs $1 \leq i < j \leq k$ (what does *more* important mean?)
   - **for** $i = 1$ **to** $k$
     - Solve MIP problem:
       $$\min_{x, y} \{c^T x + h^T y | Ax + Dy \leq b, x \in \mathbb{R}^n, 0 \leq y \leq 1, y_j \in \{0, 1\} \forall j \in Q_i\}$$
     - Let $(\bar{x}, \bar{y})$ be the solution of this MIP.
     - Fix $y_j = \bar{y}_j$ for all $j \in Q_i$, $i' \leq i$ and $j \notin Q_{i''}$, $i'' > i$

This heuristic is more suitable for problems with a “priority” structure. It also provides a lower bound at least as strong as the LPR bound (why?)

5. **Feasibility Pump** ([Fischetti et al. [2005]])
   - Let $\Delta(.)$ be a “distance” function
   - Pick a $(x^*, y^*) \in X^{LP} = \{Ax + Dy \leq b, x \in \mathbb{R}^n, 0 \leq y \leq 1\}$
   - **repeat**
     - Define the rounding $\bar{y} = |y^*|$
     - Solve $\min \{\Delta(y, \bar{y}) | (x, y) \in X\}$, obtain solution $y^*$
   - **until** $\Delta(y^*, \bar{y}) = 0$

This simple heuristic is searching the neighborhood of an LP feasible point and it is computationally proven to be useful for very hard problems by the authors. The main issue is cycling (why?), which is handled “intuitively”. 
5.2 Improvement Heuristics

Basic ideas for improvement heuristics include local search around the current best solution and exploring the sub-space spanned by the LPR solution and an MIP solution. These methods can also be extended to combining IP-feasible points if multiple solutions are available, and “mutations” are in order to diversify solution pool.

1. Local Branching (Fischetti and Lodi [2003])
   - Let $(\bar{x}, \bar{y})$ be a feasible point for MIP
   - Add the following cut to the MIP:
     \[
     \sum_{i: \bar{y}_i = 0} y_i + \sum_{i: \bar{y}_i = 1} (1 - y_i) \leq k
     \]
   - Resolve the MIP.

   This cut defines the $k$-neighborhood of $(\bar{x}, \bar{y})$. Note that this can also be used as an “exact” approach, if we branch on this cut and $\sum_{i: \bar{y}_i = 0} y_i + \sum_{i: \bar{y}_i = 1} (1 - y_i) \geq k + 1$.

2. RINS: Relaxation Induced Neighborhood Search (Danna et al. [2005])
   - Let $(\bar{x}, \bar{y})$ be an IP-feasible point and $(x^*, y^*)$ be the optimal LPR solution
   - Add the following constraints to the MIP:
     \[
     y_i = \bar{y}_i \quad \forall i \text{ with } \bar{y}_i = y^*_i
     \]
   - Resolve the MIP.

   This heuristic is based on the fact that $(x^*, y^*)$ has a good objective function value and $(\bar{x}, \bar{y})$ is integer-feasible, and exploring in between is shown to be useful.

3. Solution Crossing
   - “Crossover” idea (similar to RINS)
     - Let $(\bar{x}, \bar{y}), (x^*, y^*)$ be IP-feasible points.
     - Add the following to the MIP:
       \[
       y_i = \bar{y}_i \quad \forall i \text{ with } \bar{y}_i = y^*_i
       \]
   - “Mutation” idea
     - Let $(\bar{x}, \bar{y})$ be a feasible point for MIP
     - Let $Q \subset \{1, \ldots, p\}$ be a random set
     - Add the following to the MIP:
       \[
       y_i = \bar{y}_i \quad \forall i \in Q
       \]

   Note that these ideas need a “population” of feasible points.
5.3 Example

Let’s consider the general multi-item, multi-level production planning problem (extension of the single-item problem from the first two assignments):

\[
\begin{align*}
\min & \sum_{t=1}^{NT} \sum_{i=1}^{NI} f_t^i y_t^i + \sum_{t=1}^{NT} \sum_{i=1}^{NI} h_t^i s_t^i \\
\text{s.t.} & \quad x_t^i + s_{t-1}^i - s_t^i = d_t^i \quad t \in [1, NT], i \in [1, NI] \\
& \sum_{i=1}^{NI} (a_k^i x_t^i + ST_k^i y_t^i) \leq C_t^k \quad t \in [1, NT], k \in [1, NK] \\
& \quad x_t^i \leq M_t^i y_t^i \quad t \in [1, NT], i \in [1, NI] \\
& \quad y \in \{0, 1\}^{NT \times NI} \\
& \quad x \geq 0 \\
& \quad s \geq 0
\end{align*}
\]

When we are given a realistic problem like this, what can we do?

**Question 1**: Can we use/extend results we know for simpler models?

- Recall the \((\ell, S)\) inequalities from HW#2...
- What else do we know? Reformulations?

**Question 2**: Can we use any decomposition techniques?

- For example, consider possible Lagrangian relaxations. What can we say about these?

What about heuristic methods? Note that it is desirable to generate multiple production plans, due to the fact that production managers are picky about their plans, hence we would like to have something quick and dirty. Can we also make use of strong formulations? Let’s first make some observations:

- **Observation 1**: Decisions in earlier periods are more important than decisions in later periods (why?)
  - What does it suggest us?
- **Observation 2**: Origin is not a feasible point whereas identity vector might be feasible if capacities allow it (but it would be a bad quality solution)
  - Important on our decisions on what to fix and what not to fix

So, let’s construct a heuristic for this problem...
References


