In the following we assume all functions are linear, hence we often drop the term “linear”.

- In discrete optimization, we seek to find a solution $x^*$ in a discrete set $X$ that optimizes (maximizes or minimizes) an objective function $c^T x$ defined for all $x \in X$.

- A natural and systematic way to study a broad class of discrete optimization problems is to express them as integer programming problems. The (linear) integer programming problem is the same as the linear programming problem except that some of the variables are restricted to take integer values.

- The key concept of linear programming is that of polyhedron: a subset of $\mathbb{R}^n$ that can be described by a finite set of linear constraints $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$.

- **Def.** A polyhedron $P \subset \mathbb{R}^n$ is a valid formulation for a set $X \subset \mathbb{Z}^n$ if and only if $X = P \cap \mathbb{Z}^n$.

- Integer programming is a rather powerful modeling framework that provides great flexibility for expressing discrete optimization problems.

- On the other hand, the price for this flexibility is that integer programming is (seems to be...) a much more difficult problem than linear programming.

- We have already outlined some IP modeling techniques that facilitate the formulation of discrete optimization problems as integer programming problems.

- In comparison to linear programming, integer programming is significantly richer in modeling power. Unfortunately, there is no systematic way to formulate a discrete optimization problem, and devising a good model is often an “art”, which we plan to explore.

- In linear programming, a “good” formulation is one that has a small number $n$, $m$ of variables and constraints, respectively, because the computational complexity of the problem grows polynomially in $n$ and $m$. 
• In addition, given the availability of several efficient algorithms for linear programming, the choice of a formulation, although important, does not critically affect our ability to solve the problem.

• The situation in integer programming is drastically different. Extensive computational experience suggests that the choice of a formulation is crucial.

• In this course we provide guidelines for arriving at a “strong” integer programming formulation.

• They two key concepts of integer programming are those of linear programming relaxation and convex hull.

• Let $X \subset \mathbb{Z}^n$ be the set of feasible integer points to a particular discrete problem.

• **Def.** Given a IP valid formulation $\{x \in \mathbb{Z}^+_n \mid Ax \leq b\}$, the polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is the corresponding linear programming relaxation.

• **Def.** Given a set $S = \{x^1 \ldots x^t\}$, the convex hull of $S$, denoted $\text{conv}(S)$, is defined as $\text{conv}(S) = \{x \mid x = \sum_{i=1}^{t} \lambda_i x^i, \sum_{i=1}^{t} \lambda_i = 1, \lambda \geq 0\}$.

• In words: the convex hull of a set of points is the “smallest polyhedron” that contains all of the points in the set.

• In particular, the quality of a formulation of an integer programming problem with feasible solution set $X$, can be judged by the closeness of the feasible set of its linear programming relaxation to $\text{conv}(X)$.

• Consider two valid formulations $A$ and $B$ of the same integer programming problem. If we denote by $P_A$ and $P_B$ the feasible set of the corresponding linear programming relaxations, we consider formulation $A$ to be at least as strong as $B$ if $P_A \subseteq P_B$.

• If the inclusion is strict $P_A \subset P_B$, we say that $A$ is a “better formulation” than $B$.

• Let $z_{LP_A} = \max\{c^T x \mid x \in P_A\}$, $z_{LP_B} = \max\{c^T x \mid x \in P_A\}$, $z_{IP} = \max\{c^T x \mid x \in X\}$

• Then: $z_{IP} \leq z_{LP_A} \leq z_{LP_B}$. So, better formulations lead to “better bounds”.

• We ask: what is an ideal formulation of an integer programming problem?

• **Def.** The integral hull is the convex hull of the set of integer solutions $X$, i.e, the polyhedron $\tilde{P} = \text{conv}(X)$

• If we could represent $\text{conv}(X)$ in the form $\tilde{P} = \{x \in \mathbb{R}^n \mid \tilde{A}x \leq \tilde{b}\}$ we could solve the integer programming problem by solving a linear programming problem: $\max\{c^T x \mid x \in \tilde{P}\} = \max\{c^T x \mid x \in \text{conv}(X)\} = \max\{c^T x \mid x \in X\}$. 

2
• Motivation: all vertices of $\tilde{P} = \{ x \in \mathbb{R}^n_+ \mid \tilde{A} x \leq \tilde{b} \}$ are integer, so for any $c$ the linear program $\max \{ c^T x \mid x \in \tilde{P} \}$ has an integer optimal solution, because the optimum solution (if it exists) also occurs at a vertex which happens to be integral.

• Note that $\text{conv}(X)$ is a polyhedron if $X$ is finite, or if it is the set of feasible solutions of some MILP (not true for arbitrary set $X$!)

• Given our ability to solve linear programming problems efficiently, it is then desirable to have a formulation whose linear programming relaxation is indeed the convex hull of the integer feasible solutions $X$.

• The good news: for most polynomially-solvable COPs (e.g., assignment, min-cost flow, matching, spanning tree, etc.), we completely understand the structure of the integer hull $\tilde{P}$, so we call them “easy” polyhedra.

• We ask: do easy polyhedra have a “special” structure?

• Def. A nonempty polyhedron $P = \{ x \in \mathbb{R}^n_+ \mid Ax \leq b \}$ is said to be integral if each of its nonempty faces contains an integral point.

• Consider the linear programming problem $z_{LP} = \max \{ c^T x \mid x \in P \}$ for a given integral polyhedron $P$. The linear programming problem has an integral optimal solution for all rational vectors $c$ for which an optimal solution exists.

• Hence, if a polyhedron is integral we can optimize over it using linear programming techniques!

• We ask: under what conditions on $A$ and $b$ is it true that all vertices of $P = \{ x \in \mathbb{R}^n_+ \mid Ax \leq b \}$ happen to be integer?

• Def. An $m \times n$ integral matrix $A$ is totally unimodular (TU) if the determinant of every square sub-matrix is $0, +1, -1$.

• Theorem. (without proof) If $A$ is TU, then $P(b) = \{ x \in \mathbb{R}^n_+ \mid Ax \leq b \}$ is integral for all $b \in \mathbb{Z}^m$ (sufficiency holds).

• Theorem. (without proof) The converse is also true: if $P(b) = \{ x \in \mathbb{R}^n_+ \mid Ax \leq b \}$ is integral for all $b \in \mathbb{Z}^m$, then $A$ is TU (it also holds necessity).

• So, if the constraint matrix of an integer program is TU, it can be solved using linear programming techniques.

• We ask: how to recognize TU matrices?

• Theorem. (without proof) Here is a useful sufficient condition for $A$ to be TU.

• The rows of $A$ can be partitioned into two sets $I_1$ and $I_2$ and the following four conditions hold together:
1. Every entry $a_{ij} \in \{+1, 0, -1\}$
2. Every column of $A$ contains at most two non-zero entries.
3. If two non-zero entries in a column of $A$ have the same sign, then the row of one is in $I_1$, and the other in $I_2$.
4. If two non-zero entries in a column of $A$ have opposite sign, then the rows of both are in $I_1$, or both in $I_2$.

• The following TU properties are equivalent:
  1. $A$ is TU.
  2. The transpose of $A$ is TU.
  3. $(A|I)$ is TU.
  4. A matrix obtained by deleting a unit row/column from $A$ is TU.
  5. A matrix obtained by multiplying a row/column of $A$ by $-1$ is TU.
  6. A matrix obtained by interchanging two rows/columns of $A$ is TU.
  7. A matrix obtained by duplicating rows/columns of $A$ is TU.

• Hence, we can easily show that if $A$ is TU, it remains so after adding slack variables, and adding simple bounds on the variables.

• We can also show that the polyhedron corresponding to the dual LP is integral.

• Example: **Minimum Cost Network Flow:**
  
  – Constraint matrix: The node-arc incidence matrix of any directed graph $G = (V,E)$ is totally unimodular.
  
  – Proof: Rows have one $+1$ and one $-1$, so take $I_1 = V$ and $I_2 = \emptyset$. □
  
  – Consequence: The linear program Min-Cost Flow always has integer optimal solutions, as long as capacities $u_{ij}$ and balances $b_i$ are integer.

• Example: **Assignment Problem**
  
  – Constraint matrix arising in assignment is TU.
  
  – Proof: To see this, set $I_1 = \{1, \ldots, n\}$ and $I_2 = \{n+1, \ldots, 2n\}$. □
  
  – Thus, assignment problem can be solved by solving its LP relaxation with, e.g., Simplex method.

• So, if $A$ is TU and $b, c$ are integral vectors, then $\max\{c^T x \mid Ax \leq b, \ x \geq 0\}$ and $\min\{u^T b \mid u^T A \geq c, \ u \geq 0\}$ are attained by integral vectors $x$ and $u$ whenever the optima exist and are finite.

• This gives rise to a variety of “min-max” results.
• Example: max-flow min-cut.

• However, there are many examples where we have integral polyhedra defined by a system $Ax \leq b$ but $A$ is not TU, i.e., the polyhedron is integral only for some specific $b$.

• So we may still ask: given any $c$, consider the maximization problem $\max \{ c^T x \mid Ax \leq b, \ x \geq 0 \}$; is it the case that the dual minimization problem $\min \{ u^T b \mid u^T A \geq c, \ u \geq 0 \}$ has an integral optimal solution (whenever a finite optimum exists)?

• This motivates the following definition:

• **Def.** A rational system of inequalities $Ax \leq b$ is *totally dual integral* (TDI) if, for all integral $c$, $\min \{ u^T b \mid u^T A \geq c, \ u \geq 0 \}$ is attained by an integral vector $u^*$ whenever the optimum exists and is finite.

• Clearly, if $A$ is TU, $Ax \leq b$ is TDI for all $b$.

• This definition was introduced by Edmonds & Giles 1982, to set up the following theorem:

• **Theorem. (without proof):** If $Ax \leq b$ is TDI and $b$ is integral, then $\{ x \mid Ax \leq b \}$ is an integral polyhedron.

• This is useful because $Ax \leq b$ may be TDI even if $A$ is not TU; in other words, this is a weaker sufficient condition for integrality of $\{ x \mid Ax \leq b \}$ and moreover guarantees that the dual is integral whenever the primal objective vector is integral.

• There is an important subtlety to the definition of total dual integrality: being TDI is a property of a system of inequalities, *not* a property of the corresponding polyhedron!

• Examples: spanning tree, matching etc.

• The bad news: unfortunately, understanding the structure of the integral hull $\tilde{P}$ of NP-complete problems is often too difficult. The number of inequalities defining $\tilde{P}$ can grow very rapidly with $n$ (worse than exponential). We can’t compute them all, in fact it would be easier to solve the ILP! We call these “hard” polyhedra.

• In light of this, it is reasonable to strive for a compromise whereby one tries to come up with a polyhedron that closely *(outer-)approximates* the convex hull.

• **Def.** An inequality $\pi^T x \leq \pi_0$ is a *valid inequality* for the set $X$ if $\pi^T x \leq \pi_0$ for all $x \in X$.

• For example:
  
  - if $X = \{ x \in \mathbb{Z}^n_+ \mid Ax \leq b \}$ then all of the constraints $a_i^T x \leq b_i$ are valid inequalities for $X$. 

5
- if $\text{conv}(X) = \{ x \in \mathbb{Z}_+^n \mid \tilde{A}x \leq \tilde{b} \}$ then all of the constraints $\tilde{a}_i^T x \leq \tilde{b}_i$ are valid inequalities for $X$.

- The idea of outer-approximation is: given a valid formulation $P$ for $X$, identify additional valid inequalities for $X$ and add them to the formulation, “cutting” away regions of $P$ that contain no feasible solutions, and thus obtaining better formulations(s) for $X$.

- In fact, the main idea of the so-called cutting plane algorithm is to solve the integer programming problem by solving a sequence of linear programming problems as follows:

  - We are given an integer programming problem $\max \{ c^T x \mid x \in \mathbb{Z}_+^n, Ax \leq b \}$.
  - Let $P$ be the polyhedron associated to the linear programming relaxation: $P = \{ x \in \mathbb{R}_+^n \mid Ax \leq b \}$.
  - Clearly, the set of feasible solutions $X$ of the integer programming problem is $X = P \cap \mathbb{Z}_+^n$.
  - We first solve the linear programming relaxation $z_{LP} = \max \{ c^T x \mid x \in P \}$ and find an optimal solution $x^*$.
  - If $x^*$ is integer $\implies x^* \in X$, so it is an optimal solution to the integer programming problem.
  - If not, we find a valid inequality $\pi^T x \leq \pi_0$ (i.e., an inequality that all integer solutions in $X$ satisfy), but $x^*$ does not, i.e., $\pi^T x^* > \pi_0$ (aka violated inequality).
  - We add this inequality to $P$ to obtain a tighter relaxation $P' = \{ x \in \mathbb{R}_+^n \mid Ax \leq b, \pi^T x \leq \pi_0 \}$, and we iterate the step. (Note that: $P' \subset P$)

- We ask: how to find these inequalities?

- A simple observation: let $X = \{ x \in \mathbb{Z}_+^1 : x \leq b \}$, then the inequality $x \leq \lfloor b \rfloor$ is valid for $X$.

- The Chvátal-Gomory procedure is a systematic way of deriving inequalities for $\text{conv}(X)$ where $X = \{ x \in \mathbb{Z}_+^n \mid Ax \leq b \}$.

- Let $X = P \cap \mathbb{Z}_+^n$, where $P = \{ x \in \mathbb{R}_+^n \mid Ax \leq b \}$ and $A = [A_1 A_2 \ldots A_n] \in \mathbb{R}^{m \times n}$.

  1. Take $u \in \mathbb{R}_+^m$. The inequality $\sum_{j=1}^n u^T A_j x_j \leq u^T b$ is valid for $P$.
  2. The (weaker) inequality $\sum_{j=1}^n \lfloor u^T A_j \rfloor x_j \leq u^T b$ is also valid for $P$ (since $x \geq 0$).
  3. The inequality $\sum_{j=1}^n [u^T A_j] x_j \leq \lfloor u^T b \rfloor$ is valid for $X$.

- Every valid inequality for $X$ can be obtained by applying the Chvátal-Gomory procedure a finite number of times!
• **Gomory’s cutting-planes**: given a linear integer programming problem, the idea is to first solve the associated linear programming relaxation and find an optimal basis, choose a basic variable that is not integer, and then generate a *Gomory inequality* on the constraint associated with this basic variable so as to cut off the linear programming solution.

• Gomory’s cutting-plane algorithm is of interest because it has been shown to terminate after a finite number of iterations, but in practice it has not been successful:
  
  - A huge number of cutting planes are typically needed.
  - Cuts tend to get weaker and weaker ("tailing off").
  - Numerical errors accumulate as cuts are added.
  - No feasible integer solution is obtained until the very end.

• Also, when \( \tilde{P} = \text{conv}(X) \) is not known explicitly, checking redundancy may be very difficult!

• So now we address the question of finding *strong* valid inequalities that are hopefully even more effective, i.e., understand which are the “important” inequalities that are necessary in describing a polyhedron, and hence at least in theory provide the “best possible” cuts.

• For simplicity we limit the discussion to polyhedra \( P \subseteq \mathbb{R}^n \) that contain \( n \) linearly independent directions.

• **Def.** Such polyhedra are called *full dimensional*.

• Full dimensional polyhedra have the property that there is no equation \( a^T x = b \) satisfied at equality by all points \( x \in P \).

• **Theorem. (without proof)**: If \( P \) is a full dimensional polyhedron, it has a *unique minimal description*

\[
P = \{ x \in \mathbb{R}^n : a_i^T x \leq b_i, \ i = 1 \ldots m \}
\]

where each inequality is unique to within a positive multiple.

• **Def.**: The points \( x^1, \ldots, x^k \in \mathbb{R}^n \) are *affinely independent* if the \( k - 1 \) directions \( x^2 - x^1, \ldots, x^k - x^1 \) are linearly independent, or alternatively the \( k \) vectors \( (x^1, 1), \ldots, (x^k, 1) \in \mathbb{R}^{n+1} \) are linearly independent.

• **Def.** The *dimension* of \( P \), denoted \( \text{dim}(P) \) is one less than the maximum number of affinely independent points in \( P \).

• This means that \( P \) is full dimensional iff \( \text{dim}(P) = n \).
• **Def.**

1. $F$ defines a face of the polyhedron $P$ if $F = \{x \in P : \pi^T x = \pi_0 \}$ for some valid inequality $\pi^T x \leq \pi_0$ of $P$.
2. $F$ is a facet if $F$ is a face and $\dim(F) = \dim(P) - 1$.
3. If $F$ is a face of $P$ with $F = \{x \in P : \pi^T x = \pi_0 \}$, the valid inequality $\pi^T x \leq \pi_0$ of $P$ is said to represent or define the face.

• It follows that the faces of polyhedra are polyhedra, and it can be shown that the number of faces of a polyhedron is finite.

• Now, we establish a way to recognize the necessary inequalities.

• **Proposition. (without proof)** If $P$ is full dimensional, a valid inequality $\pi^T x \leq \pi_0$ of $P$ is necessary in the description of $P$ iff it defines a facet of $P$.

• So for full dimensional polyhedra, $\pi^T x \leq \pi_0$ of $P$ defines a facet of $P$ iff there are $n$ affinely independent points of $P$ satisfying it at equality.

• Here is a family of strong valid inequalities for 0-1 knapsack problems.

• Consider the set $X = \{x \in \{0,1\}^n : \sum_{j=1}^n a_j x_j \leq b \}$. We assume that the coefficients $\{a_j\}_{j=1}^n$ are positive and $b > 0$. Let $N = \{1 \ldots n\}$.

• **Def.:** A set $C \subseteq N$ is a cover if $\sum_{j \in C} a_j > b$.

• A cover is minimal if $C \setminus \{j\}$ is not a cover for any $j$ in $C$.

• **Proposition.:** If $C \subseteq N$ is a cover for $X$, the cover inequality

$$\sum_{j \in C} x_j \leq |C| - 1$$

is valid for $X$.

• **Proof:** We show that if $x^R$ does not satisfy the inequality, then $x^R \notin X$. If $\sum_{j \in C} x_j^R > |C| - 1$, then $|R \cap C| = |C|$ and thus $C \subseteq R$. Then, $\sum_{j=1}^n a_j x_j^R = \sum_{j \in R} a_j \geq \sum_{j \in C} a_j > b$ and so $x^R \notin X$. □

• **Proposition.:** If $C$ is a cover for $X$, the extended cover inequality

$$\sum_{j \in E(C)} x_j \leq |C| - 1$$

is valid for $X$, where $E(C) = C \cup \{j : a_j \geq a_i \ \forall i \in C\}$.

• Proof of validity as above. □
• There is one important issue that we have not discussed: how exactly do we generate cutting planes for integer programs?

• So, given a polyhedron $P$, whose integral hull is $\tilde{P}$, and a vector $x^* \in P \setminus \tilde{P}$, how do we find an inequality that is valid for $\tilde{P}$ but violated by $x^*$?

• This is called the separation problem (Grötschel, Lovász & Schrijver, 1988).

• This leads to the following concepts:
  
  – An exact separation algorithm for a given class of inequalities is a procedure which takes a vector $x^*$ as input, and either outputs one or more inequalities in the class which are violated by $x^*$, or proves that none exist.

  – A heuristic separation algorithm is similar, except that it outputs either one or more inequalities in the class which are violated by $x^*$, or a failure message.

• More formally (and general):

  – Optimization problem: given a rational polytope $P \subseteq \mathbb{R}^n$ and a rational objective vector $c \in \mathbb{R}^n$, find a $x^* \in P$ maximizing $c^T x$ over $x \in P$ or establish that $P$ is empty. (We assume that $P$ is bounded, i.e., a polytope, just to avoid unbounded problems.)

  – Separation problem: given a rational polytope $P \subseteq \mathbb{R}^n$ and a rational vector $x^*$ establish that $x^* \in P$ or determine a rational vector $(\pi, \pi_0)$ corresponding to the valid inequality $\pi^T x \leq \pi_0$ such that $\pi^T x^* > \pi_0$.

• The following important theorem in IP theory is called polynomial equivalence of separation and optimization.

• Theorem (without proof): (consequence of Grötschel, Lovász & Schrijver, 1988 theorem) The separation problem for a family of polyhedra can be solved in polynomial time in $n$ and $\log(U)$ if and only if the optimization for that family can be solved in polynomial time in $n$ and $\log(U)$, where $U$ is an upper bound on all $a_{ij}$ and $b_i$.

• In other words it says that if you can separate over $P$ in polynomial time, then you can optimize over $P$ in polynomial time, and vice-versa.

• The proof is based on the Ellipsoid method, first LP polynomial algorithm (Khachiyan, 1979).