1 Integrality of polyhedra

Consider the following LPs that we have encountered in our lectures. Recall that \( \delta(v) \) is the set of edges incident to \( v \) and \( x(F) = \sum_{e \in F} x_e \).

\[
\begin{align*}
\min c^T x & \quad \min y^T b & \quad \min p^T y \\
\forall v, x(\delta(v)) = 1 & \quad \forall (u,v) \in E, z_v = z_u + y_{u,v} & \quad \forall v, x(\delta^+(v)) - x(\delta^-(v)) = b_v \\
x \geq 0 & \quad y \geq 0 & \quad x \geq 0 \\
\end{align*}
\]

We assume that all constants involved are integers. For all these LPs, the optimal value was always integral; moreover, there existed an integral solution.

**Definition 1.1.** A polytope \( P \) is called integral if all of its vertices have integer-valued coordinates.

All these LPs led to integral polytopes, which is a bit of a surprise. There is no reason why arbitrary integral linear constraints should lead to integral vertices. The proofs of integrality were quite varied. The first polytope corresponds to min weight bipartite matching, and the integrality proof holds by Birkhoff’s theorem. The second polytope corresponds to mincut, and integrality was proven by a randomized rounding procedure. The third polytope came from min-cost flow, and the integrality proof was quite indirect. It is a consequence of Ford-Fulkerson augmenting path/cycle algorithm. The last fact is called the integrality of flow. In this lecture, we investigate a unifying theory for the integrality of these polytopes.

Another perspective to take is to think of the underlying optimization problem. In each case, we wish to optimize a linear objective over an exponential number of discrete objects. These objects are (respectively) the perfect bipartite matchings, the \( s-t \) cuts, and the integral flows satisfying the demands. Calling the set of objects \( X \), we wish to solve \( \min_{x \in X} c^T x \). We try to convert this discrete (typically \( \mathbb{NP} \)-hard) optimization problem into a continuous optimization problem as follows. Think of each element in \( X \) as an integral point in \( \mathbb{R}^m \), using the incidence vector.

**Definition 1.2.** The convex hull of a set \( X \) of points is the set of all convex combinations of \( X \). Formally \( CH(X) := \{ \sum_{x \in X} \lambda_x x \mid \lambda_x \in [0,1], \sum_x \lambda_x = 1 \} \).

Observe that we can restate our optimization problem as \( \min_{x \in CH(X)} c^T x \), which is a linear program. But this requires a polynomial description of \( CH(X) \) as a linear program (or a polynomial time separation oracle). For the settings where \( X \) is the set of perfect bipartite matchings, \( s-t \) cuts, and integral flows, the LPs above describe \( CH(X) \). Indeed, the notion of integrality of polyhedra is equivalent to the convex hull description.
Lemma 1.3. The following statements about polytope $\mathcal{P}$ are equivalent.

- $\mathcal{P}$ is integral.
- $\mathcal{P}$ is the convex hull of integral points.
- For all integral vectors $c$, $\min_{x \in \mathcal{P}} c^T x$ is an integer.

This preamble leads to us to main question of today’s lecture.

When is the polytope given by $Ax \leq b$ integral?

2 Total unimodularity

Surprisingly, we can capture a large class of integral polytopes that appears in combinatorial optimization problem. We use $\det(M)$ to denote the determinant of (square) matrix $M$.

Definition 2.1. A matrix $A$ is called totally unimodular if, for all square submatrices $A'$, $\det(A') \in \{-1, 0, 1\}$.

Note that all entries in a TU (totally unimodular) matrix lie in $\{-1, 0, 1\}$, since an individual entry is a square submatrix. We will prove the following theorem of Hoffman-Kruskal (1956).

Theorem 2.2. [Hoffman-Kruskal] Fix a matrix $A$. The polytope $\{x \mid Ax \leq b, x \geq 0\}$ is integral for all integral $b$ iff $A$ is TU.

We have a remarkable characterization of matrices that lead to integral polytopes for all $b$. From this theorem, we can also derive a weaker condition where TU leads to integrality. The following is also by Hoffman-Kruskal and is consequence of Theorem 2.2 (see exercises).

Theorem 2.3. Let $A$ be a TU matrix and $b$ be an integral vector. Then the polytope given by $Ax \leq b$ is integral.

This theorem is an important tool in combinatorial optimization. Indeed, as we show in §3, all the polytopes described in the beginning of the lecture come from TU constraint matrices. We have a grand unifying proof of the integrality of numerous polytopes that have appeared in this course.

But first, let us prove Theorem 2.2. Let us recall Cramér’s rule for solving linear equations. Let $A$ be a square, non-singular (full rank) matrix. The solution to the linear system $Ax = b$ is given by $x_i = \det(A_i)/\det(A)$, where $A_i$ is the matrix formed by replacing the $i$th column of $x$ with $b$.

Proposition 2.4. Let $A$ be a square, non-singular matrix with integral entries. Then the (unique) solution of $Ax = b$ is integral for all integral $b$ iff $|\det(A)| = 1$.

Proof. ($\Leftarrow$) If $|\det(A)| = 1$, then by Cramér’s rule, $|x_i| = |\det(A_i)|$. The latter is clearly integral for all integral $b$.

($\Rightarrow$) Note that $A^{-1}$ is the solution of the matrix equation $AX = I$ (where $I$ is the identity matrix). By assumption, the solution of $Ax = e_i$ is integral for all $i$ (where $e_i$ is the $i$th unit vector). Hence, $A^{-1}$ has all integer entries, and $\det(A^{-1})$ is an integer. But $\det(A) \cdot \det(A^{-1}) = 1$, imply that $|\det(A)| = |\det(A^{-1})| = 1$. \qed
The path to proving Theorem 2.2 will actually use a number of techniques developed to analyze the simplex algorithm. It is easier to analyze the integrality of polytopes generated by slack forms, where all constraints are either equalities or non-negativity. A polytope from a slack LP looks like \( \{ x \mid Ax = b, x \geq 0 \} \), where \( A \) can be assumed to be of full row rank. Recall that, from the lecture on the simplex algorithm, vertices in slack forms correspond to bfses. For every vertex \( v \), there exist a set of columns \( B \) such that \( A_Bv_B = b \) and all other entries of \( v \) are zero. We can combine this understanding of vertices with Prop. 2.4 to get a version of Theorem 2.2 for slack LPs.

We will first need the definition of rank unimodularity.

**Definition 2.5.** A matrix \( A \) of full row rank is called rank unimodular if, for every basis \( B \) of columns, \(| \det(A_B) | = 1 \).

Let \( A \) be an \( m \times n \) matrix of full row rank. We note that rank unimodularity is equivalent to saying that for any basis \( B \) of \( m \) columns, \( \det(A_S) = \{-1, 0, 1\} \) (Exercise). Observe that the submatrices considered in the above definition are precisely those used to define bfses. We can build on this observation to prove the following theorem.

**Theorem 2.6.** [Veinott-Dantzig 68] Fix a matrix \( A \) of full row rank. The polytope \( \{ x \mid Ax = b, x \geq 0 \} \) is integral for all integral \( b \) iff \( A \) is rank unimodular.

**Proof.** (\( \iff \)) Suppose \( A \) is rank unimodular. Consider a vertex \( v \) of the polytope \( \{ x \mid Ax = b, x \geq 0 \} \). As discussed earlier, there exists a basis \( B \) of columns such that \( A_Bv = b \). Since \( A \) is rank unimodular, \(| \det(A_B) | = 1 \). By Prop. 2.4, \( v \) is integral.

(\( \Rightarrow \)) We assume that \( \{ x \mid Ax = b, x \geq 0 \} \) is integral for all integral \( b \). Choose any basis \( B \) of columns. We need to argue that \(| \det(A_B) | = 1 \). By Prop. 2.4, it suffices to show that for all integral \( b \), the solution to \( A_Bz = b \) is integral. Consider the polytope \( \{ x \mid Ax = b, x \geq 0 \} \). If the solution to \( A_Bz = b \) is non-negative, then \( z \) (padded with extra zeroes) is a bfs and hence a vertex of this polytope. By assumption, \( z \) is integral.

Suppose the solution \( z \) has negative entries. Our strategy will be to find an integral \( b' \), such that the polytope \( \{ x \mid Ax = b', x \geq 0 \} \) has a vertex \( v \), such that \( z - v \) is integral. Observe that \( v \) is integral by assumption, so \( z \) must be integral.

Let us choose an arbitrary integral \( w \) such that \( w + z \geq 0 \), and set \( b' = A_Bw + b \). Note that \( A_B(w + z) = A_Bw + A_Bz = A_Bw + b = b' \). By construction, \( w + z \geq 0 \). Hence, \( (w + z) \) (padded with extra zeroes) is a bfs of \( \{ x \mid Ax = b', x \geq 0 \} \), and is a vertex of this polytope. So \( w + z \) is integral, implying that \( z \) is integral. \( \square \)

From here, the proof of Theorem 2.2 is fairly straightforward. We basically convert \( \{ x \mid Ax \leq b, x \geq 0 \} \) to slack form, noting that the resulting constraint matrix is rank unimodular iff \( A \) is TU.

**Proof.** (of Theorem 2.2) We start by converting the LP \( Ax \leq b, x \geq 0 \) to slack form. Recall that for every constraint \( A_i x \leq b_i \), we introduce the variable \( z_i \) and the constraints \( A_i x + z_i = b_i, z_i \geq 0 \). We get the equivalent LP

\[
[A|I] \begin{bmatrix} x \\ z \end{bmatrix} = b, \quad x, z = 0
\]

This LP/polytope is integral iff the original polytope is integral (since solutions of one LP can be mapped to the other with exactly the same objective).
Clearly, the matrix \([A|I]\) has full row rank. By Theorem 2.6, the latter polytope is integral for all \(b\) iff \([A|I]\) is rank unimodular. The matrix \(A\) is TU iff \([A|I]\) is rank unimodular, as proved in the subsequent Prop. 2.7.

**Proposition 2.7.** The matrix \(A\) is TU iff \([A|I]\) is rank unimodular.

**Proof.** \((\Rightarrow)\) Suppose the \(m \times n\) matrix \(A\) is TU. Consider the matrix \([A|I]\), which has full row rank. By the exercise after Definition 2.5, it suffices to show the following. For any subset \(S\) of \(m\) columns, \(\det([A|I]_S) \in \{-1, 0, 1\}\). If \(S\) is contained in the “\(A\)” part of the matrix, then \([A|I]_S\) is a square submatrix of \(A\). So \(\det([A|I]_S) = |\det(A')|\), for some square submatrix of \(A\). Again, since \(A\) is TU, we get the desired bound.

\((\Leftarrow)\) Suppose \([A|I]\) is rank unimodular. Let \(A'\) be a square submatrix of \(A\). Based on the columns involving \(A'\), one can add more columns from the “\(I\)” part of the matrix to get \(m\) columns \(S\) with the following property: \(\det(A') = \det([A|I]_S)\) (Exercise). The latter has value in \(\{-1, 0, 1\}\) completing the proof.

## 3 Proving the matrices at TU

To apply Theorem 2.2 or Theorem 2.3, we need convenient methods of proving that constraint matrices are TU. At the face of it, it is not clear how one would compute determinants of submatrices of, say, the mincut constraint matrix. It turns out that many of these matrices are structured enough that we can deduce that they are TU.

**Theorem 3.1.** [Poincaré 1900] Let \(A\) be a \(\{-1, 0, 1\}\) matrix. If every column has at most one 1 entry and at most one \(-1\) entry, then \(A\) is TU.

We leave the proof as an exercise. This theorem proves that one of the most common matrices in cut and flow problems, the edge incidence matrix, is TU.

Consider any digraph \(G\). The edge incidence matrix is the \(|V| \times |E|\) matrix \(B\) where:

\[
B_{v,e} = \begin{cases} 
1 & \text{if } e = (v, u) \\
-1 & \text{if } e = (u, v) \\
0 & \text{else}
\end{cases}
\]

Each column corresponds to an edge, and has exactly one 1 entry and one \(-1\) entry. As a direct corollary of Theorem 3.1, we prove that these matrices are TU.

**Corollary 3.2.** The edge incidence matrix of a digraph is TU.

Let us go back to the polytopes discussed at the beginning of the lecture and argue that they are all TU. All we need is Corollary 3.2 and a few simple exercises showing how basic modifications of a TU matrix lead to TU matrices. We will describe the matrices from these polytopes in terms of the edge incidence matrix, and leave the formal proof of being TU as an exercise.

**Mincut:** The \(s\)-\(t\) mincut constraints are the following. For every edge \((u, v)\), \(z_v - z_u - y_{u,v} \leq 0\) and \(y_{u,v} \leq 0\). (We will deal with the equalities \(z_s = 0, z_t = 1\) shortly.) These constraints lead to the matrix:

\[
\begin{bmatrix}
B^T & -I \\
0 & -I
\end{bmatrix}
\]
Note that the equalities \( z_s = 0, z_t = 1 \) reduce the set of variables (since we can set them to constants), which leads to deleting two columns of the matrix. Overall, we can construct the final matrix as follows. Starting from \( B^T \) (which is TU by Corollary 3.2), we add columns with a single non-zero entry to get \([B^T | I]\). We then add rows with single non-zero entries to get the matrix given above. Finally, we delete two columns. As described in the exercises, all these entries preserve the “TU-ness” of the matrix.

By Theorem 2.3, the s-t mincut polytope is integral. This provides an alternate proof of the max-flow-mincut theorem.

**Min-cost flow:** Max-flow is the dual of mincut, so it’s no surprise that the constraint matrix for max-flow is also TU. For the sake of completeness, let us look at the min-cost flow polytope. Observe that each row of the edge incidence matrix \( B \) is corresponds to the flow conservation constraint of a vertex. We will write the polytope as \( Ax \leq b \), hence we convert each flow conservation equality constraint into two inequalities. We have a single inequality for each capacity constraint. The constraint matrix is:

\[
\begin{bmatrix}
B \\
-B \\
-I
\end{bmatrix}
\]

We will show in the exercises that duplicating rows and negating them preserves the TU property. Since the variables are non-negative, we can apply Theorem 2.2 to deduce integrality. This gives an alternate proof of the integrality of flow, a fundamental concept in combinatorial algorithms.

**Perfect bipartite matchings:** Let us finally look at the matching polytope. Every constraint is an equality constraint \( x(\delta(v)) = 1 \), over non-negative variables. We will apply Theorem 2.6. For every \( v \) in the “left” side, we use the constraint \( x(\delta(v)) = 1 \). For every \( v \) in the “right” side, we use the negated constraint \(-x(\delta(v)) = -1\). Consider the columns of the constraint matrix, which are indexed by the variables/edges. Each column has precisely one 1 entry and one \(-1\) entry, since the graph is bipartite. Indeed, the matrix is precisely the edge incidence matrix \( B \), which we know is TU. Hence, the corresponding polytope is integral. This gives an alternate proof of Birkhoff’s theorem.

### 4 Exercises

**Easy exercises.**

1. Give an example of a polytope \( \{ x \mid Ax \leq b \} \) that is integral, but \( A \) is not TU. (Hint: Two dimensions are enough.)

2. In a rank unimodular matrix \( A \) of full row rank \( m \), prove that for any set \( S \) of \( m \) columns, \( \det(AS) = \{ -1, 0, 1 \} \).

3. Consider a matrix \( A \) of full row rank \( m \) and let \( A' \) be a submatrix. Prove that there exists a set of \( m \) columns in \([A|I] \) such that \( \det(A') = \det([A|I]_S) \).

4. Let \( A \) be TU. Prove the duplicating a row or column preserves the TU property. Also, show that negative a row or column preserves the TU property.

5. Let \( A \) be TU. Prove that adding a row or column with a single 1 entry (and all other entries being zero) preserves the TU property.

**Medium exercises.**

1. Give an example of a polytope \( \{ x \mid Ax \leq b, x \geq 0 \} \) that is integral, where \( A \) has \( \{ -1, 0, 1 \} \) values but is not TU.
2. Prove **Theorem 2.3.** (Hint: convert unconstrained variables to non-negative ones as discussed in the lecture on the simplex algorithm. Track the changes to the constraint matrix and apply **Theorem 2.2.**)

3. Prove **Theorem 3.1.** (Hint: for a submatrix, split into cases depending on whether a row/column contains a single non-zero entry.)