A \times b \text{ is integer } \implies A \times b \text{ is integral.}

\text{If } A \text{ is TD then } A \leq b \text{ is TD for all } b.

(constructor sufficient condition than TD)

(cover dual solution: \text{min } y^T \mathbf{b} : y \geq 0, \mathbf{y} = \mathbf{e}, \mathbf{y} \text{ integer})

(max \text{ c}_f \times x : A \leq b \text{ there is an integer } x \geq \text{ integer solution}

A c \geq \text{ if these is a bounded feasible solution

TD: } A \times b \leq A, b \geq \mathbf{0} \text{ (nonnegative) is TD if}

\text{unmodular: square integer matrix } (\det(c) = 1)

\text{max : vector}

\text{incidence matrix of bipartite (matching) graph is TD.}

(x \leq b) A \times b \text{ integral.}

\text{unmodular: rank } 0, \text{ not square singular submatrix}

\text{Every square non-singular submatrix is}

\text{TD: square bijective}

\text{Computational fractional values for remaining variables}

\begin{enumerate}
\item \text{Iterated rounding: Round for }
\item \text{Iterated rounding may not be the best to continue}
\item \text{Fractional solution may not be the best to continue}
\item \text{After point of the solution is rounded, the remaining}
\item \text{Does not use full power of rounding}
\item \text{Rounding based on this solution}
\item \text{Rounding: some relaxation once 8 to the}
\end{enumerate}

[unmodular] [unmodular] [unmodular]

\text{iterated elimination TD}{\text{TD of whole}}

\text{unmodular: square integer matrix (TD)}

\text{representation } [\text{Extreme point optimal solution}

\text{Combinatorial Optimization} \leftrightarrow \text{Integer Programming}
Note: For independence: (nonlinear A and convex S) if $x_i \geq 0$ for each $i$, then $\nabla f = \text{rank}(A)$.

Rank Lemma: $x_i$s are extreme point solutions if $x_i$ is an extreme point solution where $x_i < 0$.

3. Enhance extreme pt soln.

$\rightarrow$ vertex soln / basic feasible soln

$x_i$ is extreme pt soln if $f(x_0) = (\text{corner}) \leq x+y, x-y \leq f$

$p = \{ x_i \mid a_i = b, x > 0 \}$

Relaxation: Pick $x_0$: if $x_0 < \frac{a}{b}$ must. Approx. $x_0$ → $x_i = \frac{a}{b}$, $x > 0$.

Rounding: NP-hard

Optimality: inductive argument

Correctness: progress [rank lemma]

$\{ x_0 = 0 : \text{reduce convex element} \}_{\text{smaller}} \rightarrow \{ x_0 = 1 : \text{include in integral} \}_{\text{solve}}$

Iterative Alg.

Point solution:

Characterization of extreme

No solution by using separation oracle.

Exponential sized LP.
Lemma 2.1.3: \( A \) is a feasible solution to a linear programming problem if and only if it is a point in the feasible region. Let \( \mathcal{P} \) be the feasible region of the problem. Then \( \mathcal{P} \) is convex. Moreover, if \( \mathcal{P} \) is bounded, then it is compact.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is connected.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a closed set. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a complete set. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.

\[ A \rightarrow \mathcal{P} \] is a compact set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a connected set. Therefore, \( \mathcal{P} \) is complete.

\[ A \rightarrow \mathcal{P} \] is a continuous function. Therefore, \( \mathcal{P} \) is bounded.
Basic feasible solution $\iff$ extreme point solution.

And $x_B = A^B b$.

If $J \notin B$

$0 = x_J = 0 \iff A \text{ basic} \iff A \text{ is basic, is invertible.}$

Basic solution: $A$ is a subset of columns of constraint matrix $A$.

Thus any maximal $A = \text{rank}(A) = \text{maximal \# of \text{ independent } \text{ rows of } A = \text{ rank}(A) = \text{ rank}(A) = \text{ rank}(A)$

Thus any maximal $\text{rank}$ of $A$ and $\text{rank}(A)$ are equal.

Thus any maximal $\text{rank}$ of $A$ and $\text{rank}(A)$ are equal.

Thus any maximal $\text{rank}$ of $A$ and $\text{rank}(A)$ are equal.

Thus any maximal $\text{rank}$ of $A$ and $\text{rank}(A)$ are equal.

Theorem 2.1.4: $x$ is an extreme point if and only if for some $x_i \neq 0$ of $A$,

Any maximal $\# \text{ of \text{ independent } \text{ constraints } A x = b}$

Then $\# \text{ variables}$.

Thus $\# \text{ variables}$.

Theorem 2.1.4: $x$ is an extreme point if and only if for some $x_i \neq 0$ of $A$,

Then $\# \text{ variables}$.

Thus $\# \text{ variables}$.

Theorem 2.1.4: $x$ is an extreme point if and only if for some $x_i \neq 0$ of $A$, $x_i \neq 0$.

Then $\# \text{ variables}$.

Thus $\# \text{ variables}$.

Theorem 2.1.4: $x$ is an extreme point if and only if for some $x_i \neq 0$ of $A$,

Then $\# \text{ variables}$.

Thus $\# \text{ variables}$.

Theorem 2.1.4: $x$ is an extreme point if and only if for some $x_i \neq 0$ of $A$,

Then $\# \text{ variables}$.

Thus $\# \text{ variables}$.

Theorem 2.1.4: $x$ is an extreme point if and only if for some $x_i \neq 0$ of $A$,
so that \( l(v) \geq 0 \), \( x \cdot x^{\text{res}} = 0 \). \( x \cdot x = 0 \).

By induction, we get \( F \subseteq G \).

\[ \text{Proof:} \]

\[ x \text{ is feasible.} \]

\[ \text{Case 1:} \]

\[ \text{Case 2:} \]

\[ \text{Cost} \geq \text{LP} \]

\[ \text{Claim 3.1.4:} \]

\[ \text{If} \]

\[ \text{then returns a matching} \]

\[ \text{Given extreme point solution} \]

\[ \text{Lem. 3.1.3 (Converse of Rank Lemma)} \]

\[ \text{Claim 3.1.4:} \]

\[ \text{If} \]

\[ \text{Then return F} \]

\[ \text{End} \]

\[ \text{End} \]
Lemma 3.1.5: Always we get a $x = 0$ or $x = 1$.

$w(F) = w_{(F)} + w_{e} \geq w_{\text{res}} + w_{e} = w_{X}$. $w_{(F)} \geq w_{\text{res}}$.

$x \in X \Leftrightarrow \text{ a feasible solution}$

$\frac{1}{2} \leq i \leq \nu$.
\[ \text{Goal: Assign each job to some machine so that the total cost is minimized.} \]

- \( f \) = availability of machine \( \text{I} \)
- \( c_{ij} \) = cost
- \( p_i \) = processing time of job \( j \)
- \( M \) = set of machines
- \( J \) = set of jobs

A Generalized Assignment Problem:
Problem: Given a machine with $n$ processes, find the optimal schedule.

1. For each job $j$, let $t_j$ be the time it takes to complete.
2. Let $T = \sum t_j$ be the total completion time.
3. If $T > \sum t_j$, then the job cannot be completed on time. Stop.
4. Otherwise, find the optimal schedule.

Algorithm:
1. Initialize $T' = 0$.
2. For each job $j$, schedule it on the machine.
3. Update $T'$.
4. If $T' > \sum t_j$, then stop.
5. Otherwise, go to step 2.

Return $T'$ as the optimal completion time.