7.5 Bipartite Matching

Matching

- Input: undirected graph $G = (V, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a max cardinality matching.

Bipartite Matching

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a max cardinality matching.
Bipartite Matching

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a max cardinality matching.

**Max flow formulation.**
- Create digraph $G' = (L \cup R \cup \{s, t\}, E')$.
- Direct all edges from $L$ to $R$, and assign infinite (or unit) capacity.
- Add source $s$, and unit capacity edges from $s$ to each node in $L$.
- Add sink $t$, and unit capacity edges from each node in $R$ to $t$.

Bipartite Matching: Proof of Correctness

**Theorem.** Max cardinality matching in $G = \text{value of max flow in } G'$.  
**Pf.** 
- Given max matching $M$ of cardinality $k$.
- Consider flow $f$ that sends 1 unit along each of $k$ paths.
- $f$ is a flow, and has cardinality $k$.

Bipartite Matching: Proof of Correctness

**Theorem.** Max cardinality matching in $G = \text{value of max flow in } G'$.  
**Pf.** 
- Let $f$ be a max flow in $G'$ of value $k$.
- Integrality theorem $\Rightarrow k$ is integral and can assume $f$ is 0-1.
- Consider $M$ = set of edges from $L$ to $R$ with $f(e) = 1$.
  - each node in $L$ and $R$ participates in at most one edge in $M$
  - $|M| = k$: consider cut $(L \cup s, R \cup t)$.
**Perfect Matching**

**Def.** A matching $\mathcal{M} \subseteq \mathcal{E}$ is **perfect** if each node appears in exactly one edge in $\mathcal{M}$.

**Q.** When does a bipartite graph have a perfect matching?

**Structure of bipartite graphs with perfect matchings.**
- Clearly we must have $|L| = |R|$.
- What other conditions are necessary?
- What conditions are sufficient?

---

**Marriage Theorem**

**Marriage Theorem.** [Frobenius 1917, Hall 1935] Let $G = (L \cup R, \mathcal{E})$ be a bipartite graph with $|L| = |R|$. Then, $G$ has a perfect matching iff $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.

**Pf.** ⇒ This was the previous observation.

---

**Proof of Marriage Theorem**

**Pf.** ⇐ Suppose $G$ does not have a perfect matching.
- Formulate as a max flow problem and let $(A, B)$ be min cut in $G'$.
- By max-flow min-cut, $\text{cap}(A, B) < |L|$.
- Define $L_A = L \cap A$, $L_B = L \cap B$, $R_A = R \cap A$.
- $\text{cap}(A, B) = |L_B| + |R_A|$.
- Since min cut can’t use $\infty$ edges: $N(L_A) \subseteq R_A$.
- $|N(L_A)| = |R_A| = \text{cap}(A, B) - |L_B| < |L| - |L_B| = |L_A|$.
- Choose $S = L_A$. □

---

**Notation.** Let $S$ be a subset of nodes, and let $N(S)$ be the set of nodes adjacent to nodes in $S$.

**Observation.** If a bipartite graph $G = (L \cup R, \mathcal{E})$, has a perfect matching, then $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.

**Pf.** Each node in $S$ has to be matched to a different node in $N(S)$.
Which max flow algorithm to use for bipartite matching?

- Generic augmenting path: $O(m \text{val}(f^*)) = O(mn)$.
- Capacity scaling: $O(m^2 \log C) = O(m^2)$.
- Shortest augmenting path: $O(m n^{1/2})$.

Non-bipartite matching.

- Structure of non-bipartite graphs is more complicated, but well-understood. [Tutte-Berge, Edmonds-Galai]
- Blossom algorithm: $O(n^6)$. [Edmonds 1965]
- Best known: $O(m n^{1/2})$. [Micali-Vazirani 1980]

### 7.6 Disjoint Paths

**Disjoint path problem.** Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find the max number of edge-disjoint $s$-$t$ paths.

**Def.** Two paths are edge-disjoint if they have no edge in common.

**Ex:** communication networks.
**Max flow formulation:** assign unit capacity to every edge.

**Theorem.** Max number edge-disjoint s-t paths equals max flow value.

**Pf. ≤**
- Suppose there are k edge-disjoint paths $P_1, \ldots, P_k$.
- Set $f(e) = 1$ if $e$ participates in some path $P_i$; else set $f(e) = 0$.
- Since paths are edge-disjoint, $f$ is a flow of value $k$. ▪

**Network Connectivity**

**Network connectivity.** Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find min number of edges whose removal disconnects $t$ from $s$.

**Def.** A set of edges $F \subseteq E$ disconnects $t$ from $s$ if every s-t path uses at least one edge in $F$.

**Theorem.** [Menger 1927] The max number of edge-disjoint s-t paths is equal to the min number of edges whose removal disconnects t from s.

**Pf. ≤**
- Suppose the removal of $F \subseteq E$ disconnects $t$ from $s$, and $|F| = k$.
- Every s-t path uses at least one edge in $F$.
- Hence, the number of edge-disjoint paths is at most $k$. ▪

---

**Diagram 1:**
- Edge Disjoint Paths
- Max flow formulation: assign unit capacity to every edge.
- Theorem. Max number edge-disjoint s-t paths equals max flow value.
- Pf. ≤
  - Suppose there are k edge-disjoint paths $P_1, \ldots, P_k$.
  - Set $f(e) = 1$ if $e$ participates in some path $P_i$; else set $f(e) = 0$.
  - Since paths are edge-disjoint, $f$ is a flow of value $k$. ▪

**Diagram 2:**
- Edge Disjoint Paths and Network Connectivity
- Network connectivity. Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find min number of edges whose removal disconnects $t$ from $s$.
- Def. A set of edges $F \subseteq E$ disconnects $t$ from $s$ if every s-t path uses at least one edge in $F$.
- Theorem. [Menger 1927] The max number of edge-disjoint s-t paths is equal to the min number of edges whose removal disconnects $t$ from $s$.
- Pf. ≤
  - Suppose the removal of $F \subseteq E$ disconnects $t$ from $s$, and $|F| = k$.
  - Every s-t path uses at least one edge in $F$.
  - Hence, the number of edge-disjoint paths is at most $k$. ▪
Disjoint Paths and Network Connectivity

**Theorem.** [Menger 1927] The max number of edge-disjoint s–t paths is equal to the min number of edges whose removal disconnects t from s.

**Pf.**
- Suppose max number of edge-disjoint paths is \( k \).
- Then max flow value is \( k \).
- Max-flow min-cut \( \Rightarrow \) cut \( (A, B) \) of capacity \( k \).
- Let \( F \) be set of edges going from \( A \) to \( B \).
- \(|F| = k\) and disconnects \( t \) from \( s \).

---

7.7 Extensions to Max Flow

**Circulation with Demands.**

- Directed graph \( G = (V, E) \).
- Edge capacities \( c(e), e \in E \).
- Node supply and demands \( d(v), v \in V \).

\[
\text{demand if } d(v) > 0; \text{ supply if } d(v) < 0; \text{ transshipment if } d(v) = 0
\]

**Def.** A **circulation** is a function that satisfies:

- For each \( e \in E \): \( 0 \leq f(e) \leq c(e) \) (capacity)
- For each \( v \in V \): \( \sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v) \) (conservation)

**Circulation problem:** given \( (V, E, c, d) \), does there exist a circulation?

---

**Necessary condition:** sum of supplies = sum of demands.

\[
\sum_{v : d(v) > 0} d(v) = \sum_{v : d(v) < 0} d(v) =: D
\]

**Pf.** Sum conservation constraints for every demand node \( v \).
Max flow formulation.

\begin{itemize}
\item Add new source \( s \) and sink \( t \).
\item For each \( v \) with \( d(v) < 0 \), add edge \( (s, v) \) with capacity \(-d(v)\).
\item For each \( v \) with \( d(v) > 0 \), add edge \( (v, t) \) with capacity \( d(v) \).
\item Claim: \( G \) has circulation iff \( G' \) has max flow of value \( D \).
\end{itemize}

**Integrality theorem.** If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.

**Pf.** Follows from max flow formulation and integrality theorem for max flow.

**Characterization.** Given \((V, E, c, d)\), there does not exist a circulation iff there exists a node partition \((A, B)\) such that \(\sum_{v \in B} d_v > \text{cap}(A, B)\).

**Pf idea.** Look at min cut in \( G' \).

---

**Feasible circulation.**

- Directed graph \( G = (V, E) \).
- Edge capacities \( c(e) \) and lower bounds \( \ell(e) \), \( e \in E \).
- Node supply and demands \( d(v) \), \( v \in V \).

**Def.** A circulation is a function that satisfies:

- For each \( e \in E \): \( \ell(e) \leq f(e) \leq c(e) \) (capacity)
- For each \( v \in V \): \( \sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v) \) (conservation)

**Circulation problem with lower bounds.** Given \((V, E, \ell, c, d)\), does there exist a circulation?
Circulation with Demands and Lower Bounds

**Idea.** Model lower bounds with demands.
- Send \( l(e) \) units of flow along edge \( e \).
- Update demands of both endpoints.

**Theorem.** There exists a circulation in \( G \) iff there exists a circulation in \( G' \). If all demands, capacities, and lower bounds in \( G \) are integers, then there is a circulation in \( G \) that is integer-valued.

**Pf sketch.** \( f(e) \) is a circulation in \( G \) iff \( f'(e) = f(e) - l(e) \) is a circulation in \( G' \).

### 7.8 Survey Design

**Survey Design**

**Survey design.**
- Design survey asking \( n_1 \) consumers about \( n_2 \) products.
- Can only survey consumer \( i \) about product \( j \) if they own it.
- Ask consumer \( i \) between \( c_i \) and \( c_i' \) questions.
- Ask between \( p_j \) and \( p_j' \) consumers about product \( j \).

**Goal.** Design a survey that meets these specs, if possible.

**Bipartite perfect matching.** Special case when \( c_i = c_i' = p_i = p_i' = 1 \).

**Algorithm.** Formulate as a circulation problem with lower bounds.
- Include an edge \( (i, j) \) if consumer \( j \) owns product \( i \).
- Integer circulation \( \iff \) feasible survey design.
Image Segmentation

7.10 Image Segmentation

- Central problem in image processing.
- Divide image into coherent regions.

Ex: Three people standing in front of complex background scene. Identify each person as a coherent object.

Formulate as min cut problem.
- Maximization.
- No source or sink.
- Undirected graph.

Turn into minimization problem.
- Maximizing is equivalent to minimizing

Goals.
- Accuracy: if \( a_i > b_i \) in isolation, prefer to label \( i \) in foreground.
- Smoothness: if many neighbors of \( i \) are labeled foreground, we should be inclined to label \( i \) as foreground.
- Find partition \((A, B)\) that maximizes:
  \[
  \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij} \]
  
  or alternatively
  \[
  \sum_{j \in B} b_j + \sum_{i \in A} a_i - \sum_{(i,j) \in E} p_{ij} \]

Foreground / background segmentation.
- Label each pixel in picture as belonging to foreground or background.
- \( V \) = set of pixels, \( E \) = pairs of neighboring pixels.
- \( a_i \geq 0 \) is likelihood pixel \( i \) in foreground.
- \( b_i \geq 0 \) is likelihood pixel \( i \) in background.
- \( p_{ij} \geq 0 \) is separation penalty for labeling one of \( i \) and \( j \) as foreground, and the other as background.

Goals.
- Accuracy: if \( a_i > b_i \) in isolation, prefer to label \( i \) in foreground.
- Smoothness: if many neighbors of \( i \) are labeled foreground, we should be inclined to label \( i \) as foreground.
- Find partition \((A, B)\) that maximizes:

\[
\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij}
\]

\[
\sum_{j \in B} b_j + \sum_{i \in A} a_i - \sum_{(i,j) \in E} p_{ij} \]

\[
\sum_{(i,j) \in E} a_i \cdot \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij} \cdot \sum_{i \in A} a_i + \sum_{j \in B} b_j \]

\[
\sum_{j \in B} b_j + \sum_{i \in A} a_i - \sum_{(i,j) \in E} p_{ij} \]

\[
\sum_{(i,j) \in E} a_i \cdot \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij} \cdot \sum_{i \in A} a_i + \sum_{j \in B} b_j \]

\[
\sum_{(i,j) \in E} a_i \cdot \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij} \cdot \sum_{i \in A} a_i + \sum_{j \in B} b_j \]

\[
\sum_{j \in B} b_j + \sum_{i \in A} a_i - \sum_{(i,j) \in E} p_{ij} \]

\[
\sum_{(i,j) \in E} a_i \cdot \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij} \cdot \sum_{i \in A} a_i + \sum_{j \in B} b_j \]

\[
\sum_{j \in B} b_j + \sum_{i \in A} a_i - \sum_{(i,j) \in E} p_{ij} \]

\[
\sum_{(i,j) \in E} a_i \cdot \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij} \cdot \sum_{i \in A} a_i + \sum_{j \in B} b_j \]

\[
\sum_{j \in B} b_j + \sum_{i \in A} a_i - \sum_{(i,j) \in E} p_{ij} \]
Formulate as min cut problem.

- \( G' = (V', E') \).
- Add source to correspond to foreground; add sink to correspond to background.
- Use two anti-parallel edges instead of undirected edge.

Consider min cut \((A, B)\) in \(G'\).

- \( A = \text{foreground} \).
- Precisely the quantity we want to minimize.

\[
\text{cap}(A, B) = \sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{(i, j) \in E, i \in A, j \in B} p_{ij} \quad \text{if } i \text{ and } j \text{ on different sides, } p_{ij} \text{ counted exactly once}
\]

7.11 Project Selection

Projects with prerequisites.

- Set \( P \) of possible projects. Project \( v \) has associated revenue \( p_v \).
  - some projects generate money: create interactive e-commerce interface, redesign web page
  - others cost money: upgrade computers, get site license
- Set of prerequisites \( E \). If \((v, w) \in E\), can’t do project \( v \) and unless also do project \( w \).
- A subset of projects \( A \subseteq P \) is feasible if the prerequisite of every project in \( A \) also belongs to \( A \).

Project selection. Choose a feasible subset of projects to maximize revenue.
Prerequisite-graph.
- Include an edge from $v$ to $w$ if can’t do $v$ without also doing $w$.
- $\{v, w, x\}$ is feasible subset of projects.
- $\{v, x\}$ is infeasible subset of projects.

Min cut formulation.
- Assign capacity $\infty$ to all prerequisite edge.
- Add edge $(s, v)$ with capacity $p_v$ if $p_v > 0$.
- Add edge $(v, t)$ with capacity $-p_v$ if $p_v < 0$.
- For notational convenience, define $p_s = p_t = 0$.

Claim. $(A, B)$ is min cut iff $A - \{s\}$ is optimal set of projects.
- Infinite capacity edges ensure $A - \{s\}$ is feasible.
- Max revenue because: $\text{cap}(A, B) = \sum_{v \in B: p_v > 0} p_v - \sum_{v \in A: p_v < 0} p_v$.

Open-pit mining. (studied since early 1960s)
- Blocks of earth are extracted from surface to retrieve ore.
- Each block $v$ has net value $p_v =$ value of ore - processing cost.
- Can’t remove block $v$ before $w$ or $x$. 
7.12 Baseball Elimination

"See that thing in the paper last week about Einstein? . . . Some reporter asked him to figure out the mathematics of the pennant race. You know, one team wins so many of their remaining games, the other teams win this number or that number. What are the myriad possibilities? Who’s got the edge?"

"The hell does he know?"

"Apparently not much. He picked the Dodgers to eliminate the Giants last Friday."

- Don DeLillo, Underworld

Which teams have a chance of finishing the season with most wins?

- Montreal eliminated since it can finish with at most 80 wins, but Atlanta already has 83.
- \( w_i + r_i < w_j \Rightarrow \text{team } i \text{ eliminated.} \)
- Only reason sports writers appear to be aware of.
- Sufficient, but not necessary!

<table>
<thead>
<tr>
<th>Team</th>
<th>Wins ( w_i )</th>
<th>Losses ( l_i )</th>
<th>To play ( r_i )</th>
<th>Against = ( r_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atlanta</td>
<td>83</td>
<td>71</td>
<td>8</td>
<td>- 1 6 1</td>
</tr>
<tr>
<td>Philly</td>
<td>80</td>
<td>79</td>
<td>3</td>
<td>1 - 0 2</td>
</tr>
<tr>
<td>New York</td>
<td>78</td>
<td>78</td>
<td>6</td>
<td>6 0 - 0</td>
</tr>
<tr>
<td>Montreal</td>
<td>77</td>
<td>82</td>
<td>3</td>
<td>1 2 0 -</td>
</tr>
</tbody>
</table>

Remark. Answer depends not just on how many games already won and left to play, but also on whom they’re against.
Baseball Elimination

Baseball elimination problem.
- Set of teams $S$.
- Distinguished team $s \in S$.
- Team $x$ has won $w_x$ games already.
- Teams $x$ and $y$ play each other $r_{xy}$ additional times.
- Is there any outcome of the remaining games in which team $s$ finishes with the most (or tied for the most) wins?

Can team 3 finish with most wins?
- Assume team 3 wins all remaining games $\Rightarrow w_3 + r_3$ wins.
- Divvy remaining games so that all teams have $\leq w_3 + r_3$ wins.

Theorem. Team 3 is not eliminated iff max flow saturates all edges leaving source.
- Integrality theorem $\Rightarrow$ each remaining game between $x$ and $y$ added to number of wins for team $x$ or team $y$.
- Capacity on $(x, t)$ edges ensure no team wins too many games.

Baseball Elimination: Explanation for Sports Writers

<table>
<thead>
<tr>
<th>Team</th>
<th>Wins $w_i$</th>
<th>Losses $l_i$</th>
<th>To play $r_i$</th>
<th>Against $= r_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>75</td>
<td>59</td>
<td>28</td>
<td>NY 3 Baltimore 8 7 3</td>
</tr>
<tr>
<td>Baltimore</td>
<td>71</td>
<td>63</td>
<td>28</td>
<td>3 2 7 4</td>
</tr>
<tr>
<td>Boston</td>
<td>69</td>
<td>66</td>
<td>27</td>
<td>8 2 - 0 0</td>
</tr>
<tr>
<td>Toronto</td>
<td>63</td>
<td>72</td>
<td>27</td>
<td>7 7 0 - -</td>
</tr>
<tr>
<td>Detroit</td>
<td>49</td>
<td>86</td>
<td>27</td>
<td>3 4 0 0 -</td>
</tr>
</tbody>
</table>

**Which teams have a chance of finishing the season with most wins?**
- Detroit could finish season with $49 + 27 = 76$ wins.
Which teams have a chance of finishing the season with most wins?
- Detroit could finish season with 49 + 27 = 76 wins.

Certificate of elimination. R = {NY, Bal, Bos, Tor}
- Have already won w(R) = 278 games.
- Must win at least \( r(R) = 27 \) more.
- Average team in R wins at least \( 305/4 > 76 \) games.

Certificate of elimination.

<table>
<thead>
<tr>
<th>Team</th>
<th>Wins</th>
<th>Losses</th>
<th>To play</th>
<th>Against = ( r_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>75</td>
<td>59</td>
<td>28</td>
<td>- 3 8 7 3</td>
</tr>
<tr>
<td>Baltimore</td>
<td>71</td>
<td>63</td>
<td>28</td>
<td>3 - 2 7 4</td>
</tr>
<tr>
<td>Boston</td>
<td>69</td>
<td>66</td>
<td>27</td>
<td>8 2 - 0 0</td>
</tr>
<tr>
<td>Toronto</td>
<td>63</td>
<td>72</td>
<td>27</td>
<td>7 7 0 - -</td>
</tr>
<tr>
<td>Detroit</td>
<td>49</td>
<td>86</td>
<td>27</td>
<td>3 4 0 0 - -</td>
</tr>
</tbody>
</table>

Baseball Elimination: Explanation for Sports Writers

Certificate of elimination.

\[
T \subseteq S, \quad w(T) := \sum_{i \in T} w_i, \quad g(T) := \sum_{(x, y) \in T} (w_x + g_y),
\]

LB on avg \# games won

If \( \frac{w(T) + g(T)}{\# T} > w_z + g_z \) then z is eliminated (by subset T).

Theorem. [Hoffman-Rivlin 1967] Team z is eliminated iff there exists a subset \( T^* \) that eliminates z.

Proof idea. Let \( T^* \) = team nodes on source side of min cut.
k-Regular Bipartite Graphs

Dancing problem.
- Exclusive Ivy league party attended by \( n \) men and \( n \) women.
- Each man knows exactly \( k \) women; each woman knows exactly \( k \) men.
- Acquaintances are mutual.
- Is it possible to arrange a dance so that each woman dances with a different man that she knows?

Mathematical reformulation. Does every \( k \)-regular bipartite graph have a perfect matching?

Ex. Boolean hypercube.

Census Tabulation (Exercise 7.39)

Feasible matrix rounding.
- Given a \( p \)-by-\( q \) matrix \( D = \{d_{ij}\} \) of real numbers.
- Rows \( i \) sum = \( a_i \), column \( j \) sum = \( b_j \).
- Round each \( d_{ij}, \ a_i, \ b_j \) up or down to integer so that sum of rounded elements in each row (column) equals row (column) sum.
- Original application: publishing US Census data.

Goal. Find a feasible rounding, if one exists.

| 1.4 | 6.2 | 7.3 | 17.24 |
| 3.6 | 2.4 | 0.7 | 12.7  |
| 3.6 | 1.2 | 6.5 | 11.3  |
| 16.34| 10.4| 14.5|

original matrix

| 3  | 7  | 7  | 17  |
| 10 | 2  | 1  | 13  |
| 3  | 1  | 7  | 11  |
| 16 | 10 | 15 | 11  |
feasible rounding
Feasible matrix rounding.
- Given a p-by-q matrix \( D = \{d_{ij}\} \) of real numbers.
- Row i sum = \( a_i \), column j sum = \( b_j \).
- Round each \( d_{ij}, a_i, b_j \) up or down to integer so that sum of rounded elements in each row (column) equals row (column) sum.
- Original application: publishing US Census data.

**Goal.** Find a feasible rounding, if one exists.

**Remark.** “Threshold rounding” can fail.

\[
\begin{array}{cccc}
0.35 & 0.35 & 0.35 & 1.05 \\
0.55 & 0.55 & 0.55 & 1.65 \\
0.9 & 0.9 & 0.9 & \\
\end{array}
\]

original matrix

\[
\begin{array}{ccc}
0 & 0 & 1 \\
1 & 1 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

feasible rounding

**Theorem.** Feasible matrix rounding always exists.

**Pf.** Formulate as a circulation problem with lower bounds.
- Original data provides circulation (all demands = 0).
- Integrality theorem \( \Rightarrow \) integral solution \( \Rightarrow \) feasible rounding.

\[
\begin{array}{cccc}
3.14 & 6.8 & 7.3 & 17.24 \\
9.6 & 2.4 & 0.7 & 12.7 \\
3.6 & 1.2 & 6.5 & 11.3 \\
16.34 & 10.4 & 14.5 & \\
\end{array}
\]

lower bound

upper bound

\[
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
0 & ∞ & \\
\end{array}
\]

row

column