Maximum Weighted Matching

Ran Duan
In this lecture

• Hall’s theorem
• Maximum weighted bipartite matching
• Hungarian algorithm
Hall’s Theorem

- Given a bipartite graph $G=(L \cup R, E)$, where $|L|=|R|$, 
  - It contains a perfect matching if and only if:
    - For every subset $S \subseteq L$, $|\Gamma(S)| \geq |S|$
    - ($\Gamma(S)$ is the set of vertices adjacent to $S$)
Proof on Induction

- Let $n = |L| = |R|$
- When $n = 1$, trivial
- Suppose it holds for all $n \leq k$, for $n = k + 1$, two cases:
Proof on Induction

- Let \( n = |L| = |R| \)
- When \( n = 1 \), trivial
- Suppose it holds for all \( n \leq k \), for \( n = k + 1 \), two cases:
  - Case I: For every subset \( S \subseteq L \), \( |\Gamma(S)| \geq |S| + 1 \)
    - Then we arbitrarily put an edge \((u,v)\) in the matching
    - In \( G - \{u,v\} \), it still satisfies the condition \( |\Gamma(S)| \geq |S| \), so the result holds by the induction condition
Proof on Induction

• Case II: there exists a $T \subseteq L$ which has $|\Gamma(T)| = |T|$, then the subgraphs of $G$ on:
  ▫ $T \cup \Gamma(T)$
  ▫ $(L \setminus T) \cup (R \setminus \Gamma(T))$
both satisfies the Hall’s condition
Proof on Induction

- **Case II:** there exists a $T \subseteq L$ which has $|\Gamma(T)| = |T|$, then the subgraphs of $G$ on:
  - $T \cup \Gamma(T)$
  - $(L \setminus T) \cup (R \setminus \Gamma(T))$

both satisfies the Hall’s condition

There may be an edge between $L \setminus T$ and $\Gamma(T)$
But there are no edge between $T$ and $R \setminus \Gamma(T)$
Proof on Induction

- In $T \cup \Gamma(T)$, every $S \subseteq T$ have $\Gamma(S) \subseteq \Gamma(T)$, so it satisfies the Hall’s condition
Proof on Induction

- In \((L \setminus T) \cup (R \setminus \Gamma(T))\), if \(\exists S \subseteq L \setminus T\) having \(|\Gamma(S) \cap (R \setminus \Gamma(T))| < |S|\), then \(T \cup S\) will also break the Hall’s condition for \(G\), a contradiction.

So \((L \setminus T) \cup (R \setminus \Gamma(T))\) satisfies the Hall’s condition.
Proof on Induction

- Case II: there exists a $T \subseteq L$ which has $|\Gamma(T)| = |T|$, then the two subgraphs of $G$ on:
  - $T \cup \Gamma(T)$
  - $(L \setminus T) \cup (R \setminus \Gamma(T))$
  both satisfies the Hall’s condition

So we can find perfect matchings in these two subgraphs, and finally get a perfect matching of $G$. 
Weighted Bipartite Matching

- Maximum Weighted Matching (MWM)
  - Maximize $\sum_{e \in M} w(e)$
Assignment Problem

- In operation research:
  - Some agents, some tasks
  - Assign each task to an agent
  - Maximize efficiency or minimize cost

### Assignment Problem

<table>
<thead>
<tr>
<th></th>
<th>Cleaning</th>
<th>Sweeping</th>
<th>Washing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim</td>
<td>$2</td>
<td>$6</td>
<td>$4</td>
</tr>
<tr>
<td>Steve</td>
<td>$3</td>
<td>$1</td>
<td>$2</td>
</tr>
<tr>
<td>Alan</td>
<td>$5</td>
<td>$4</td>
<td>$3</td>
</tr>
</tbody>
</table>
Weighted Bipartite Matching

- When not every pair of vertices of L and R has an edge, we can consider two problems:
  - Maximum (Minimum) perfect matching
    - The maximum or minimum among all perfect matchings
  - Maximum matching
    - Not necessarily perfect
Weighted Bipartite Matching

- When not every pair of vertices of L and R has an edge, we can consider two problems:
  - Maximum (Minimum) perfect matching (MWPM)
    ▫ The maximum or minimum among all perfect matchings
  - Maximum matching (MWM)
    ▫ Not necessarily perfect
Reduction between MWM and MWPM

- **MWM=>MWPM**
  - We add zero-weight edge for any pair of \((u,v)\) if there is no edge between \((u,v)\). \((u \in L, v \in R)\)
  - In the new graph any matching can be extend to a perfect matching of the same weight, so the maximum perfect matching must have maximum weight.
Reduction between MWM and MWPM

- **MWM=>MWPM**
  - We add zero-weight edge for any pair of \((u,v)\) if there is no edge between \((u,v)\). \((u \in L, v \in R)\)
  - In the new graph any matching can be extend to a perfect matching of the same weight, so the maximum perfect matching must have maximum weight.

- It will increase the number of edges
Reduction between MWM and MWPM

- **MWM=>MWPM**
  - Duplicate $G$, we have $G_1=(L_1,R_1)$ and $G_2=(L_2,R_2)$.
  - Link the two copies of every vertex of $G$ by an edge with weight zero.
  - Still a bipartite graph: one side $L_1 \cup R_2$, the other side $L_2 \cup R_1$. 
Reduction between MWM and MWPM

- **MWM => MWPM**
  - Duplicate $G$, we have $G_1 = (L_1, R_1)$ and $G_2 = (L_2, R_2)$.
  - Link the two copies of every vertex of $G$ by an edge with weight zero.
  - Still a bipartite graph: one side $L_1 \cup R_2$, the other side $L_2 \cup R_1$. 

Reduction between MWM and MWPM

- **MWM => MWPM**
  - Duplicate $G$, we have $G_1 = (L_1, R_1)$ and $G_2 = (L_2, R_2)$.
  - Link the two copies of every vertex of $G$ by an edge with weight zero.
  - Still a bipartite graph: one side $L_1 \cup R_2$, the other side $L_2 \cup R_1$.
Reduction between MWM and MWPM

- **MWM=>MWPM**
  - Duplicate G, we have $G_1=(L_1,R_1)$ and $G_2=(L_2,R_2)$.
  - Link the two copies of every vertex of G by an edge with weight zero.
  - Still a bipartite graph: one side $L_1 \cup R_2$, the other side $L_2 \cup R_1$.
  - The number of vertices and edges are still $O(n)$ and $O(m)$, respectively.
Reduction between MWM and MWPM

- **MWPM => MWM**
  - If the weights are in $[0, ..., N]$, add $nN$ to the weight of every edge, and get a new graph $G'$
  - The weight of a matching of $k$ edges in $G'$ is $\leq k(n+1)N$ (when $k \leq n-1$, $k(n+1)N < n^2N$)
  - The weight of a perfect matching in $G'$ is $\geq n^2N$
  - So the maximum matching in $G'$ must be a perfect matching.
Hungarian Algorithm

• By Harold Kuhn in 1955, who gave the name because it was largely based on the earlier works of two Hungarian mathematicians: Dénes Kőnig and Jenő Egerváry.
• In 2006, it was discovered that Carl Jacobi had solved the assignment problem in the 19th century.
Hungarian Algorithm

• By Harold Kuhn in 1955, who gave the name because it was largely based on the earlier works of two Hungarian mathematicians: Dénes Kőnig and Jenő Egerváry.

• In 2006, it was discovered that Carl Jacobi had solved the assignment problem in the 19th century.

• We will first talk about the maximum perfect matching.
Hungarian Algorithm

- Dual variable $y$: $L \cup R \rightarrow \mathbb{Z}$ satisfies:
  - For every $e=(u,v)$, $y(u)+y(v) \geq w(e)$
Hungarian Algorithm

- Dual variable $y: \mathbb{L} \cup \mathbb{R} \to \mathbb{Z}$ satisfies:
  - For every $e = (u, v)$, $y(u) + y(v) \geq w(e)$

- So for every perfect matching $M$,

$$w(M) = \sum_{e \in M} w(e) \leq \sum_{v \in \mathbb{L} \cap \mathbb{R}} y(v)$$
Hungarian Algorithm

- Dual variable $y: \mathbb{L} \cup \mathbb{R} \rightarrow \mathbb{Z}$ satisfies:
  - For every $e=(u,v)$, $y(u) + y(v) \geq w(e)$

- So for every perfect matching $M$,
  \[
  w(M) = \sum_{e \in M} w(e) \leq \sum_{v \in L \cap R} y(v)
  \]

- Our aim: obtain a perfect matching $M^*$ s.t.
  - for every $e \in M^*$, $y(u) + y(v) = w(e)$
Throughout the algorithm:

- $y(u) + y(v) \geq w(e) \quad \forall \ e = (u,v)$ (domination)
- $y(u) + y(v) = w(e) \quad \text{if } e \in M$ (tightness)
Throughout the algorithm:
- $y(u) + y(v) \geq w(e)$ $\forall e = (u,v)$ (domination)
- $y(u) + y(v) = w(e)$ if $e \in M$ (tightness)

Tight edges:
- An edge $e = (u,v)$ is tight if $y(u) + y(v) = w(e)$
- Denote the subgraph of tight edges by $G_y$
Procedure

- Let \( y(u) = N, \ y(v) = 0 \) (\( u \in L, \ v \in R \))
- Repeat
  - Augment \( M \) in \( G_y \) (subgraph of tight edges), until there is no augmenting path any more.
  - If \( M \) is not perfect, do the dual adjustment to make more edges tight.
- Until \( M \) is perfect
Procedure

- Let \( y(u) = N, y(v) = 0 \) \((u \in L, v \in R)\)
- Repeat
  - Augment \( M \) in \( G_y \) (subgraph of tight edges), until there is no augmenting path any more. (Augmentation step)
  - If \( M \) is not perfect, adjust the dual variable \( y \) to make more edges tight. (Dual adjustment step)
- Until \( M \) is perfect
Augmentation step

- Find $G_y$ (subgraph of tight edges)
  - From the tightness condition, all matching edges are in $G_y$
- Finding augmenting path as in cardinality matching
- Until there is no augmenting paths any more.
Augmentation step

• An example:
Augmentation step

• An example:
Augmentation Step

- We can use breadth-first search to find augmenting paths.
- It takes $O(m)$ time for one path.
Dual-adjustment step

- We assign directions to edges in $G_y$ and get $G_y'$:
  - Non-matching edges: from L to R
  - Matching edges from R to L
  - A path between free vertices of L and R in $G_y'$ $\Leftrightarrow$ An augmenting path in $G_y$
Dual-adjustment step

- We assign directions to edges in $G_y$ and get $G_y'$:
  - Non-matching edges: from L to R
  - Matching edges from R to L
  - A path between free vertices of L and R in $G_y'$ $\Leftrightarrow$ An augmenting path in $G_y$

- We have to guarantee there is no augmenting path in $G_y$ before the dual-adjustment
- So there is no directed path between free vertices of L and R in $G_y'$
An example

$G_y$

$G'_y$
Dual-adjustment

- In $G'_y$, find the vertices reachable from free vertices of L, call this set $Z$
  - Since there is no directed path between free vertices of L and R in $G'_y$, $Z$ does not contain free vertices of R
Dual-adjustment

- In $G_y'$, find the vertices reachable from free vertices of $L$, call this set $Z$
  - Since there is no directed path between free vertices of $L$ and $R$ in $G_y'$, $Z$ does not contain free vertices of $R$

- Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
- Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
  - $\Delta$ can bring more tight edges without breaking the domination condition
  - For integer-weighted graph, we can set $\Delta = 1$
An example

- Tight edges
- Matching edges

(Dual adjustment step)

Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
An example

- Tight edges
- Matching edges

(Dual adjustment step)

Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
Correctness

• Z is the set of vertices reachable from free vertices of L
• all vertices in L-Z are matched
• all vertices in R∩Z are matched
Correctness

- Z is the set of vertices reachable from free vertices of L

- for a matching edge \((u,v)\), either:
  - \(u\) and \(v\) are both in \(Z\)
  - \(u\) and \(v\) are neither in \(Z\)
  - (If \(v\) is in \(Z\), \(u\) must be in \(Z\))
  - (\(u\) can only be reached from \(v\))
Correctness

• $Z$ is the set of vertices reachable from free vertices of $L$

• for a matching edge $(u,v)$, either:
  ▫ $u$ and $v$ are both in $Z$
  ▫ $u$ and $v$ are neither in $Z$

• So after the dual-adjustment, all matching edges still satisfy $y(u)+y(v)=w(u,v)$
For non-matching edges

- $Z$ is the set of vertices reachable from free vertices of $L$ by tight edges
- There is no tight edges $(u,v)$ from $L \cap Z$ to $R - Z$
  - Otherwise $v$ will be in $Z$
For non-matching edges

- $Z$ is the set of vertices reachable from free vertices of $L$ by tight edges
- There is no tight edges $(u,v)$ from $L \cap Z$ to $R - Z$

- For edges $(u,v)$ from $L - Z$ to $R \cap Z$
  - Only $v$ increase
  - The domination condition $y(u) + y(v) \geq w(u,v)$ still holds
For non-matching edges

- $Z$ is the set of vertices reachable from free vertices of $L$ by tight edges.
- There is no tight edges $(u,v)$ from $L \cap Z$ to $R - Z$.
- So the amount of adjustment
  $$\Delta = \min\{y(u) + y(v) - w(u,v) \mid u \in L \cap Z, v \in R - Z\}$$
  - So we can have more tight edges, and $Z$ will get larger.
For non-matching edges

- So the amount of adjustment
  \[ \Delta = \min \{ w(u,v) - y(u) - y(v) \mid u \in L \cap Z, v \in R - Z \} \]
  - So we can have more tight edges, and \( Z \) will get larger.
  - Until some free vertex is added to \( Z \)
Let \( y(u) = N, \ y(v) = 0 \) (\( u \in L, \ v \in R \))

Repeat

- Augment \( M \) in \( G_y \) (subgraph of tight edges), until there is no augmenting path any more. (Augmentation step)

- If \( M \) is not perfect, adjust the dual variable \( y \) to make more edges tight. (Dual adjustment step)
  - Let \( Z \) be the set of vertices reachable from free vertices of \( L \)
  - Let \( y(u) = y(u) - \Delta \) for \( u \in L \cap Z \)
  - Let \( y(v) = y(v) + \Delta \) for \( v \in R \cap Z \)

Until \( M \) is perfect
Running Time

- M can be augmented \( n \) times
- There can be at most \( O(n) \) dual-adjustment steps before M can be augmented
  - Every time Z becomes larger
- The time needed by searching for an augmenting path or a dual-adjustment step is \( O(m) \)
- The total time is \( O(mn^2) \)
An example
An example

- Tight edges

(Augmenting step)
find augmenting path

Diagram showing a network with labeled edges and nodes.
An example

- Tight edges
- Matching edges
An example

- Tight edges
- Matching edges

(Dual adjustment step)

Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
An example

- Tight edges
- Matching edges

(Dual adjustment step)
Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
An example

- Tight edges
- Matching edges

(Augmenting step)
find augmenting path
An example

- Tight edges
- Matching edges
An example

- Tight edges
- Matching edges

(Dual adjustment step)
Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
An example

- Tight edges
- Matching edges

(Dual adjustment step)
Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
An example

- Tight edges
- Matching edges

(Dual adjustment step)
Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u)=y(u)-\Delta$ for $u \in L \cap Z$
Let $y(v)=y(v)+\Delta$ for $v \in R \cap Z$
An example

- Tight edges
- Matching edges

(Dual adjustment step)
Let $Z$ be the set of vertices reachable from free vertices of $L$
Let $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
Let $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
An example

- Tight edges
- Matching edges

(Augmenting step)
find augmenting path
Finally

- Note that y-value can be negative
Termination condition

- If we want a **maximum (minimum) perfect matching**, then we stop when we get a perfect matching $M^*$.

- Now $w(M^*) = \sum_{e \in M^*} w(e) = \sum_{v \in L \cap R} y(v)$.

- For every other perfect $M$, $w(M) = \sum_{e \in M} w(e) \leq \sum_{v \in L \cap R} y(v)$.

- So $w(M^*) \geq w(M)$.
Termination condition

- If we want a **maximum matching**, then we stop when the free vertices of L have zero y-value.

  - The y-value of free vertices are decreased by the same amount in every step, so they remain equal throughout the algorithm.
Termination condition

- Since at the beginning, $y(L) = N$, $y(R) = 0$
- In the dual-adjustment step:
  - $y(u) = y(u) - \Delta$ for $u \in L \cap Z$
  - $y(v) = y(v) + \Delta$ for $v \in R \cap Z$
- $Z$ does not contain free vertices in $R$, otherwise there will be augmenting paths
- So the free vertices of $R$ have zero $y$-value throughout the algorithm
Termination condition

• If we want a maximum matching, then we stop when the free vertices of L have zero y-value, and get $M^*$
• Then all free vertices have zero y-value.

Now
$$w(M^*) = \sum_{e \in M^*} w(e) = \sum_{v \in L \cap R} y(v)$$

For every other $M$,  
$$w(M) = \sum_{e \in M} w(e) \leq \sum_{v \in L \cap R} y(v)$$

So $w(M^*) \geq w(M)$
In the example

For maximum perfect matching

For maximum matching
Approximate matching (optional)

• Add a little relaxation on the tightness condition
• Converge more quickly
Original conditions

- Throughout the algorithm:
  - $y(e) \geq w(e)$ (domination)
  - $y(e) = w(e)$ if $e \in M$ (tightness)
Relaxed conditions

- Throughout the algorithm:
  - $y(e) \geq w(e) - \frac{1}{k}$ (domination)
  - $y(e) = w(e)$ if $e \in M$ (tightness)
Relaxed conditions

• Throughout the algorithm:
  ▫ $y(e) \geq w(e) - \frac{1}{k}$ (domination)
  ▫ $y(e) = w(e)$ if $e \in M$ (tightness)

• Then we run the Hungarian search on eligible edges:
  ▫ $y(e) = w(e) - \frac{1}{k}$ if $e$ not in $M$
  ▫ all the matching edges
• After augmentation, we add 1/k to the R-side vertex of every new matching edges, so the tightness for matching edges still holds.
• After augmentation, we add $1/k$ to the R-side vertex of every new matching edges, so the tightness for matching edges still holds.
• So other edges associated with these vertices will not be eligible any more
• After augmentation, we add $1/k$ to the R-side vertex of every new matching edges, so the tightness for matching edges still holds.
• So other edges associated with these vertices will not be eligible any more
• We just need to find a maximal set of augmenting paths in $O(m)$ time, then there will be no augmenting path before dual-adjustment
• After $kN$ dual-adjustments we can get a $(1-1/k)$-approximate maximum weighted matching
About the exam time

• All students are now asked to register in HISPOS for the exams for the summer term 2012.

• Please inform the students about the obligatory examination registration.

• In case of problems with the registration, the students can send an email to
  ▪ studium@cs.uni-saarland.de
Next lecture

- Maximum weighted matching in general graphs
- Some applications of matching