Matching in Bipartite Graphs

Arash Rafiey

17 November 2015
We have a bipartite graph \( G = (C, R, E) \) where \( R \) represents a set of resources and \( C \) represents a set of customers.

The edge set shows a customer in \( C \) likes (willing to have) a subset of resources in \( R \) (not necessary all of them).
We have a bipartite graph $G = (C, R, E)$ where $R$ represents a set of resources and $C$ represents a set of customers.

The edge set shows a customer in $C$ likes (willing to have) a subset of resources in $R$ (not necessary all of them).

Our goal is to assign (some of ) the resources to customers such that each customer receives one resource from its neighborhood.

In other words, we would like to match each customer with a resource.
A matching in graph $G$ is the set $M \subseteq E$ s.t for every $e, f \in M$, $e, f$ do not have a node in common.

$M_1 = \{c_1r_1, c_2r_4, c_3r_2\}$

$M_1$ is a matching.
\[ M_2 = \{ c_1r_1, c_2r_1, c_3r_2 \} \]

\( M_2 \) is not a matching.
We say a matching $M$ is **perfect** (with respect to $C$) if it contains all the vertices in $C$.

We say a matching $M$ is **maximum** (with respect to $C$) if $|C \cap M|$ is maximum (has maximum number of edges).
Hall’s condition

If there is a set of customer $C'$ such that in their neighborhood there is not enough resources. Then there is no perfect matching.

In the Figure $|C'| = 3$, $N(C') = \{r_2, r_3\}$ and $|C'| > |N(C')|$. Therefore there is no perfect matching.
If the Hall’s condition satisfies for every subset of $C$ then there is a perfect matching in $G$.

$$\forall C' \subseteq C, |C'| \leq |N(C')|$$

$N(C')$ is the neighborhood of $C'$. All the elements that have at least one neighbor in $C'$. 

Arash Rafiey
Matching in Bipartite Graphs
An exchanging and expanding idea

Suppose we have a matching $M$ and it has $c_1 r_1$ and $c_2 r_2$. Suppose $c_3 r_2$, $c_2 r_1$, $c_1 r_3$ are edges of $G$.

Then we can start from $c_3$ and go to $r_2$ then to $c_2$ and then to $r_1$ and then to $c_1$ and then to $r_3$.

We can replace $r_1 c_1$ and $r_2 c_2$ by $c_3 r_2$, $c_2 r_1$, $c_1 r_3$.

\[
M = \{c_1 r_1, c_2 r_2\}
\]
An exchanging and expanding idea

Suppose we have a matching $M$ and it has $c_1r_1$ and $c_2r_2$. Suppose $c_3r_2$, $c_2r_1$, $c_1r_3$ are edges of $G$.

Then we can start from $c_3$ and go to $r_2$ then to $c_2$ and then to $r_1$ and then to $c_1$ and then to $r_3$.

We can replace $r_1c_1$ and $r_2c_2$ by $c_3r_2$, $c_2r_1$, $c_1r_3$.

\[ M = \{c_1r_1, c_2r_2\} \]
\[ M_{\text{new}} = \{c_3r_2, c_2r_1, c_1r_3\} \]
Let $M$ be a matching. We say a vertex is \textit{matched} if it belongs to an edge $e \in M$. 
Let $M$ be a matching. We say a vertex is matched if it belongs to an edge $e \in M$.

We start with an unmatched node $u$ and find an alternating path $P$ from $u$ to another unmatched node $v$ such that:

First edge of $P$ is not in $M$ and second edge of $P$ is in $M$ and the third edge of $P$ is not in $M$ and so on. The last edge of $P$ is not in $M$.

If $P$ exists then we set $M' = (P - M) \cup (M - P)$. $M'$ has one edge more than $M$. 
Let $M$ be a matching. We say a vertex is matched if it belongs to an edge $e \in M$.

We start with an unmatched node $u$ and find an alternating path $P$ from $u$ to another unmatched node $v$ such that:

First edge of $P$ is not in $M$ and second edge of $P$ is in $M$ and the third edge of $P$ is not in $M$ and so on. The last edge of $P$ is not in $M$.

If $P$ exists then we set $M' = (P - M) \cup (M - P)$. $M'$ has one edge more than $M$. 
Matching( G=(C,R,E) )

1) Start with a matching with one edge. As long as you can add edge into it (without violating the definition)

2) Start with a vertex \( v \) in \( C \) that is not matched in \( M \)

3) Find an alternating path \( P \) form \( v \) to some other un matched vertex \( u \) (use modified version of DFS algorithm)

4) If \( P \) exists then set \( M' = (P - M) \cup (M - P) \) and go to (2)

5) Else there is no perfect matching
Matching( \( G=(C,R,E) \) )

1) Start with a matching with one edge. As long as you can add an edge into it (without violating the definition)

2) Start with a vertex \( v \) in \( C \) that is not matched in \( M \)

3) Find an alternating path \( P \) form \( v \) to some other unmatched vertex \( u \) (use modified version of DFS algorithm)

4) If \( P \) exists then set \( M' = (P - M) \cup (M - P) \) and go to (2)

5) Else there is no perfect matching

Running time \( \mathcal{O}(nm) \) (\( m \) is the number of edges) when we use link list. The reason is we consider every unmatched vertex \( v \) (there are at most \( n \) nodes) and for that vertex we run DFS with \( \mathcal{O}(m) \).
Matching in Bipartite Graphs
Matching in Bipartite Graphs
Apply $M = (M - P) \cup (P - M)$
Start with unmatched 3
$P = 3, b, 2, a, 1, c$

Apply $M = (M - P) \cup (P - M)$
Start with unmatched 4

\[ P = 4, b, 3, a, 2, e, 5, d \]
Start with unmatched 4

\[ P = 4, b, 3, a, 2, e, 5, d \]

Apply

\[ M = (P - M) \cup (M - P) \]
M is perfect
Matching(G=(C,R,E))

0. Set $M = \emptyset$.

1. for (every edge $uv \in E$)
2.     if ($u \not\in M$ and $v \not\in M$)
3.         $M = M \cup \{uv\}$.

4. for (every $u \in C$ where $u \not\in M$)
5.     FOUND=0; $P = \emptyset$;
6.     Au-DFS(u,1);
7.     if (FOUND==1)
8.         $M = (M - P) \cup (P - M)$
Au-DFS \((u, \text{flag})\)

1. \(\textbf{if } (\text{flag}==2 \text{ and } u \not\in M)\)
2. add \(u\) to \(P\) and set FOUND=1; // a path found
3. exit;

5. Explore\([u]\)=1; add \(u\) to \(P\).

6. \(\textbf{if } (\text{flag}==1)\)

7. \(\textbf{for } (\text{every } w \in N(u) \text{ and } uw \not\in M )\)
8. \(\textbf{if } (\text{Explore}[w]==0 )\)
9. \(\text{Au-DFS}(w,2)\);
10. remove \(w\) from \(P\).

11. \(\textbf{if } (\text{flag}==2)\)

12. \(\textbf{for } (\text{every } w \in N(u) \text{ and } uw \in M)\)
13. \(\textbf{if } (\text{Explore}[w]==0 )\)
14. \(\text{Au-DFS}(w,1)\);
15. remove \(w\) from \(P\).
Theorem

Bipartite graph $G = (C, R, E)$ has a perfect matching with respect to $C$ if and only if the Hall’s condition is satisfied, i.e.
$\forall X \subseteq C, |X| \leq |N(X)|$.

Proof:
If there exists a perfect matching in $G$ w.r.t. $C$ then clearly for each subset $C'$ of $C$, we have $|C'| \leq |N(C')|$. Now suppose the Hall’s condition satisfied for every subset $C'$ of $C$. We show that at each step of the algorithm there exists an augmenting (alternating) path. Suppose there exits no augmenting path $P$ for unmatched vertex $u$ with partial matching $M$. 
Au-DFS algorithm explores a tree $T$ from node $u$. $u$ is on level 0. All the neighbors of $u$ (on level 1) are matched with a unique edge in $M$ otherwise there exists $P$. The vertices on level 3 are also matched by new edges in $M$. In general, for each new node $w$ on an odd level of $T$ there exists a new edge in $M$ that matches $w$. Since there exists no path $P$, the number of nodes on even levels is less than the number of nodes on odd levels. Set $C'$ be the set of vertices on even levels of $T$ and $N(C')$ be the vertices of odd levels of $T$. As argued, we have $|C'| > |N(C')|$, a contradiction.
1) We are given a $k$-regular bipartite graph $H$ (every vertex has degree $k$). Show that we can color the edges of $H$ with $k$-colors such that the edges incident to a same vertex receive different colors.
1) We are given a $k$-regular bipartite graph $H$ (every vertex has degree $k$). Show that we can color the edges of $H$ with $k$-colors such that the edges incident to a same vertex receive different colors.

2) We are given a bipartite graph $H = (C, R, E)$. We want to find an assignment of the resources to the customers such that each customer receives at least $b$ resources from $R$. What is the necessary condition?