CO650

Rui Gong

February 25, 2022

Contents

1 Minimum-cost Spanning Trees

What's a spanning tree?

Definition 1

Given a graph $G = (V, E)$, a subgraph T is a spanning tree of G if:

- $V(T) = V(G)$
- T is connected
- T is acyclic (contains no cycle)

Theorem 2

Let $G = (V, E)$ be connected graph, T be a subgraph of G, with $V(T) = V$, then the following are equivalent (TFAE)

- T is a spanning tree of G.
- T is minimally connected (T will be disconnected if any edge is dropped).
- T is maximally acyclic (Add any edge between vertices of T makes it cyclic).
- $\forall u, v \in V$, there exists a unique $u - v$ path in T (call it $T_{u,v}$).

Theorem 3

A graph $G = (V, E)$ is connected if and only if $\forall A \subseteq V$ with $\emptyset \neq A \neq V$, we have $\delta(A) \neq \emptyset$ ($\delta(A) := \{e \in E : |e \cap A| = 1\}$, the set of edges with exactly one edge in A).

1.1 Minimum Spanning Tree Problem

Input:

- Connected graph $G = (V, E)$.
- Costs $C_e, \forall e \in E$.

Output: A spanning tree T of G of minimum cost $C(T) := \sum_{e \in E(t)} C_e$.

Theorem 4

Let $G = (V, E)$, connected, $C : E \mapsto \mathbb{R}$, T is a spanning tree of G, then TFAE:

a) T is a MST (minimum spanning tree).

b) $\forall uv \in E \setminus E(T)$, all edges e on $T_{u,v}$ have $C_e \leqslant C_{uv}$.

c) $\forall e \in E(T)$, let T_1, T_2 be the two connected components obtained from T when removing e. then e is a min cost edge in $\delta(T_1)$ (of G).

Proof.

- $a) \implies b)$. Suppose $\exists uv \in E \setminus E(T)$ and $e \in T_{u,v}$ such that $C_e > C_{uv}$, consider $T' = T + uv - e$. Since we don't change delete any vertices, $V(T') = V(T) = V(G)$. If we write $T_{u,v} = u, v_1, \ldots, v_n, v$ and say v_i, v_{i+1} are the ends of e. Then, for any two vertices of G, if they are not on $T_{u,v}$, they are still connected. If at least one of them is on $T_{u,v}$, say k_2 is on $T_{u,v}$, WLOG, say the unique path is $k_1, \ldots, u, \ldots, k_2$, if e is not in the path, we are good, if it is, then we take $k_1, \ldots, u, v, \ldots, k_2$ on T'. Hence, T' is connected. That is, T' is connected and $|E(T')| = |E(T)| = |V| - 1$, so by theorem 2, we know T' is a spanning tree of G. And since $C_{uv} < C_e$, $C(T') < C(T)$, T is not a MST, contradiction, so no such uv exists.

- $b) \implies c)$. Suppose $\exists e \in T, uv \in \delta(T_1)$ such that $C_{uv} < C_e$. First, $uv \notin E(T)$, otherwise, since $v \in T_2$ and T_1, T_2 are connected, so there is a cycle including uv and e in T, contradiction. Also, $e \in T_{uv}$, because we have $u \in T_1$, and $v \in T_2$, and T_1, T_2 are separated by e, so any path from T_1 to T_2 will include e. Then this contradicts to $b)$, contradiction, no such e exists, $c)$ is true.

- $c) \implies a)$. Let T satisfy $c)$. Let T^* be a MST with largest $k := |E(T) \cap E(T^*)|$.
If $k = n - 1 = |V| - 1$, we are done.
Else, there is $e \in E(T) \setminus E(T^*)$ (note T is also a spanning tree). Let T_1, T_2 be connected component of $T - e$, there exists $e^* \in E(T^*) \cap \delta(T_1)$.
First $e^* \notin E(T)$ because otherwise, we have e and e^* connecting T_1 and T_2 in T (note $e \neq e^*$).
Also, $T' = T^* - e^* + e$ is also a spanning tree, because all vertices stay connected and the number of edges stay the same (as above proof).
By $c)$, $C_e \leqslant C_{e^*}$, so $C(T) \leqslant C(T^*)$. So T' is a MST, and $|E(T) \cap E(T')| = k + 1 > k$, contradiction. So $k = n - 1$, $T = T^*$ which is a MST.

$\square$

1.2 Kruskal's Algorithm

Algorithm 1 Kruskal's Algorithm for MST

Require: G be a connected graph, $n = |V|$, $m = |E|$
  $H = (V, \emptyset)$
  while H is not a spanning tree **do**
    Find the cheapest edge whose endpoints are in different components of H
    $H \leftarrow H + e$
  end while

We also have an equivalent version:

Algorithm 2 Equivalent

  Sort edges so that $C_{e_1} \leqslant \ldots \leqslant C_{e_m}$.
  for $i = 1, \ldots, m$ **do**
    if endpoints u, v of e_i are in different components of H **then**
      $H \leftarrow H + e_i$
    end if
  end for

Implementation:

- Keep array $comp$, with $comp[v] \leftarrow v, \forall v \in V$ initially.
- The if step in algorithm 2 can be done by checking if $comp[u] == comp[v]$, for $e = uv$. $O(1)$.
- When the assignment step in alg2 is executed, go through $comp[t], \forall t \in V$, if $comp[t] == comp[v]$, set $comp[t] = comp[u]$. That is, make sure u, v and the vertices they are connected to are in the same component. $O(n)$.
- Sort step $O(m \log m)$.
- For loop step $O(m)$ in total.

Overall, we have $O(m \log m) + O(mn) = O(mn)$, which is a polynomial time. At the end, H will be a spanning tree.

Q: Can alg1 get stuck?

- The e we need to find always exists. Since H is not a spanning tree, either it is not connected, or it has a cycle in it. However, if H has a cycle, the last edge added to that cycle will not be added because its two endpoints are already in H. Hence, H is disconnected, so we can find an edge connecting different components of H.
- Everytime $H \leftarrow H + e$ is executed, two different components are connected, so the number of different connected components of H minus 1. Also, since H is acyclic before the assignment, and e connects two different components, there is no cycle.

- Every iteration, the number of components minus 1, and we have n components at the beginning, so we do $O(n)$ iterations, during the time, we keep the H acyclic.

Q: Does it return a MST?
Suppose not, there exists $uv \in E \setminus E(H)$ and $e \in H_{uv}$ with $C_{uv} < C_e$ by theorem4, so C_{uv} will be considered before C_e in the first step of alg2. When C_{uv} is being tested in alg2's third step, there is no path from u to v, so they are in different components, so uv will be added to H, not e, contradiction.

1.3 Correctness via LP

- Show techniques that can be used in other settings
- Lead to "good" approaches for more challenging problems

1.3.1 Integer Programming Formulation

- Let $x_e \in \{0, 1\}$ to indicate if edge e is in the MST
- Spanning tree: acylic, $n - 1$ edges ($n := |v|$)

$$\begin{aligned}
&\min \sum_{e \in E} c_e x_e \ c^T x \\
&s.t.\ x(E) = n - 1 \text{ ,where } x(F) := \sum_{e \in F} x_e \\
&\quad x(F) \leqslant n - \kappa(F), \forall F \subseteq E \\
&\quad x \in \{0, 1\}^E
\end{aligned}$$

For **Acyclic:**
Consider $F \subseteq E$. How many edges of F can a spanning tree have?
Let $\kappa(F)$ be the number of connected components of (V, F), then our answer is $n - \kappa(F)$. Since if we consider every connected components of (V, F), we can find a spanning tree in it and have at most the number of vertices in that component minus one edges. So, sum over all components, we have $n - k(F)$ at most without forming a cycle.
Note: If $F = \{e\}$, then $\kappa(F) = n - 1$, so $x(F) \leqslant n - \kappa(F)$ becomes $x_e \leqslant 1$, so our problem becomes

$$\begin{aligned}
&\min \sum_{e \in E} c_e x_e \\
&s.t.\ x(E) = n - 1 \\
&\quad x(F) \leqslant n - \kappa(F), \forall F \subseteq E \\
&\quad x \geqslant 0, x \in \mathbb{Z}^E
\end{aligned}$$

1.3.2 LP relaxation:

$$\begin{aligned}(P_{ST}),\ \zeta^*_{P_{ST}} := \min\ & \sum_{e\in E} c_e x_e \\ s.t.\ & x(E) = n-1 \\ & x(F) \leqslant n - \kappa(F), \forall F \subseteq E \\ & x \geqslant 0\end{aligned}$$

It has optimal solutions. Since G is connected, it has feasible solutions (just find a spanning tree) and its feasible regions is bounded, so it has optimal solutions.

Proof Idea:

- Any spanning tree T corresponds to a feasible solution to $(P_{ST}) \implies c(T) \geqslant \zeta^*_{P_{ST}}$.
- Shows that spanning tree produced by Kruskal is optimal for (P_{ST}) (using Complementary Slackness).

$$\begin{aligned}\min\ & c^T x \\ s.t.\ & x(E) = n-1 \\ & x(F) \leqslant n - \kappa(F), \forall F \subseteq E \\ & x \geqslant 0\end{aligned}$$

note that $n - 1 = n - \kappa(E)$. Then we find the dual

Dual (D_{ST}):

$$\begin{aligned}\max\ & \sum_{F\subseteq E} (n-\kappa(F)) y_F \\ s.t.\ & \sum_{F: e\in F} y_F \leqslant c_e, \forall e \in E \\ & y_F \leqslant 0, \forall F \subset E \\ & y_E \text{ is free.}\end{aligned}$$

Let $E = \{e_1, \ldots, e_m\}$, with $c_{e_1} \leqslant c_{e_2} \leqslant \ldots \leqslant c_{e_m}$. Let $E_i = \{e_1, \ldots, e_i\}$,

- $\overline{y}_{E_i} = c_{e_i} - c_{e_{i+1}} \leqslant 0, \forall i = 1, \ldots, m-1$
- $\overline{y}_E = c_{e_m}, \overline{y}_F = 0, \forall$ other F.

Now, we want to show that $\overline{y}$ is feasible for (D_{ST}) and all constraints are satisfied at equality (except the $y_F \leqslant 0$ ones). For each $e_i \in E$, we know $e_i \in E_i, \ldots, E_m$ and some other non-E_i edge subsets. Hence,

$$\begin{aligned}\sum_{F: e_i\in F} y_F &= \sum_{j=i}^{m} y_{E_j} + \sum_{F\neq E_j, j\geqslant i: e_i\in F} y_F \\ &= c_{e_i} - c_{e_m} + c_{e_m} + 0 \\ &= c_{e_i}\end{aligned}$$

So the Complementary Slackness condition for Dual constraints are satisfied, we only need to check either $\overline{y}_i = 0$ or $\overline{x}_{E_i}$ constraint is tight.

Now, let $\overline{x}$ be the incidence vector of tree T constructed by Krustal. Note: $\overline{x}(E_i) = \sum_{e \in E_i} \overline{x}_e = |E(T) \cap E_i|$.

- $T_i = (V, E_i \cap E(T))$ is a maximally acyclic subgraph of $H_i = (V, E_i)$. Suppose not, then we can add an edge e_k of $E_i \setminus E(T)$ to T_i, and it's still acyclic. This edge e_k connects two component of T_i, otherwise, since $e_k \notin E(T)$, its endpoints are in the same component in T, so there is a path between its endpoints in $E_k \cap E(T) \subseteq E_i \cap E(T)$, contradiction. Since it connects two components of T_i, it will added be to T at k^{th} iteration, so it will be in $E(T)$, contradiction.
- As argued before, $n - \kappa(E_i)$ is the largest number of edges we can choose from E_i without forming a cycle in $H_i = (V, E_i)$, that is, by previous point, $n - \kappa(E_i) = |E_i \cap E(T)| = \overline{x}(E_i)$.
- Now we argue the Complementary Slackness conditions are satisfied. For each $F \subseteq E$, if $F \in \{E_1, \ldots, E_m\}$, then by the previous point, the equality is tight; otherwise, $y_F = 0$. For each $e \in E$, we showed that all constraints of the dual problem are tight. So $\overline{x}, \overline{y}$ are optimal for P_{ST}, D_{ST} respectively.
- Hence, $c^T \overline{x} = c(T) = \zeta^*_{P_{ST}}$ by Complementary Slackness Theorem.

Consequence of Proof:

- $\zeta^*_{P_{ST}} = c(T^*)$, where T^* is MST.
- Solving the above LP can give us an integral solution (under mild assumptions), which rarely happens.

Alternative Formulation for P_{ST}:

$$
\begin{aligned}
\zeta^*_{P_{ST}} := \min\ & c^T x \\
s.t.\ & x(E) = n - 1 \\
& x(E(S)) \leqslant |S| - 1, \forall \emptyset \subsetneq S \subsetneq V \\
& x \geqslant 0
\end{aligned}
$$

where $E(S) = \{e \in E : |e \cap S| = 2\}$.

1.3.3 Greedy and Max Cost Forest

- MST Algorithms are greedy (best decision based only on local structure).
- Ex: Max weight independent set. Given $G = (V, E)$, $S \subseteq V$ is an independent set if $\forall u, v \in S$, $uv \notin E$. Then given C_v, $\forall v \in V$, find independent set S, which maximize $c(S) := \sum_{v \in S} c_v$.

Maximum Forest Problem:
Given $G = (V, E)$, a forest is a subgraph (V, F) with $F \subseteq E$ that is acyclic. (We refer to a forest by its set of edges). Then we want

Given $G = (V, E)$, $c_e, \forall e \in E$, find a forest F maximizing $c(F) := \sum_{e \in F} c_e$.

USE MST:

- Compute MST with respect to $c'_e = -c_e$.
- Delete from MST all edges with $c_e \leqslant 0$.

Remark. If G is not connected, add edges to it with cost $-M$, where $M > 0$ is large.

- The above algorithm will compute a max cost forest: Consider any two components of the computed forest. By the definition of MST algorithm, the edge deleted from the computed spanning tree has the smallest cost in the edges between the two components (w.r.t. $-c_e$), so all edges between this two components have negative costs. Also, for any edges e not connecting two components of the forest, if it has a positive cost, then it will be added to the computed spanning tree, hence a contradiction. Similar for the case when an edge is between two vertices of a component of the forest.
- We should have M greater than the largest absolute value of the negative c_e, so that when we are computing the MST, the "added" edges will never be selected.

1.3.4 Kruskal for Maximum Cost Forest

Algorithm 3

```
H = (V, ∅).
while ∃e : c_e > 0, with endpoints in different connected components of H do
    e = highest cost edge whose endpoints are in different components of H.
    H ← H + e
end while
    return H
```

To solve MST (alternatively):

- Add $-M$ to $c_e, \forall e$ such that $c_e - M < 0$
- Solve maximum cost forest w.r.t. $c'_e = -(c_e - M)$

If G is connect, and with all costs $c'_e > 0$, the above algorithm will find a spanning tree with the largest cost w.r.t. c'_e, that is, a spanning tree with the smallest cost w.r.t c_e.

2 Matroids

Look at edge setes of forests, i.e. instead of finding $H = (V, F)$, we just refer to F.

Algorithm 4 Generic Greedy

$F \leftarrow \emptyset$.
while $\exists e : F \cup \{e\} \in I$ and $c_e > 0$ **do**
  choose such e with largest c_e;
  $F \leftarrow F \cup \{e\}$
end while
  return F

where I here represents the set of all forests.

2.1 Matroid 1

Definition 5: Matroids

Let S be a ground set. Let $I \subseteq 2^S$ (the set of all subsets of S). $M = (S, I)$ is called a Matroid if it satisfies the following:

(M1) $\emptyset \in I$

(M2) If $F \in I$, $F' \subseteq F$, then $F' \in I$.

(M3) For all $A \subseteq S$, every inclusionwise maximal element of I that is contained in A (definition of the basis of A) has the same cardinality. That is, $B \in I$ is a subset of A and no other subsets of A in I is a strict superset of B, then B is a basis of A.

Example 6

- Let $G = (V, E)$. Set $S = E, I = \{$ all forest$\}$. We get a **Graphical/Forest Matroid**.
- Let $S = \{1, \ldots, n\}$. Let $r \in \{0, \ldots, n\}$, $I =$ set of all subsets of S with at most r elements. We have

$$U_n^r = (S, I) \implies \textbf{Uniform matroid of rank } r$$

Q1: Is U_n^r a matroid?

(M1) $|\phi| = 0 \leqslant r$

(M2) If $A \in I$ and $B \subseteq A$, then $|B| \leqslant |A| \leqslant r \implies B \in I$.

(M3) If there are two basis B_1, B_2 of A and $|B_1| < |B_2| \leqslant \min\{r, |A|\}$, then let $e \in B_2 \setminus B_1$. Then $B_1 \cup \{e\} \subseteq A$ and $|B_1 \cup \{e\}| \leqslant |B_2| \leqslant \min\{r, |A|\}$. So B_1 is not a basis, contradiction.

- Let N be an $m \times n$ matrix of real numbers. Let $S = \{1, \ldots, n\}$. $I = \{A \subseteq S :$ columns indexed by A are linearly independent.$\}$. We call this a **Linear Matroid**. e.g.:

$$N = \begin{pmatrix} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \end{pmatrix}$$

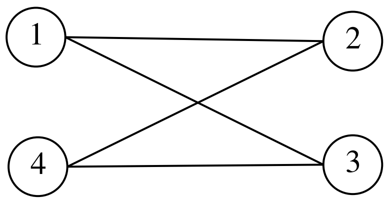
, then $I = \{\emptyset, \{1\}, \ldots, \{5\}, \{1, 2\}, \{1, 3\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \ldots\}$.

(M1) $\emptyset \in I$.

(M2) If a set of vectors is linearly independent, then any subset of it is also linearly independent.

(M3) Follows from linear algebra. (Note: basis of a vector space transalte to basis of A).

Example 7



Let $S = V$, $I = \{A \subseteq V : A \text{ is a stable set}\}$.

(M1) $\emptyset \in I$

(M2) A subset of a stable set is also stable set.

(M3) Maximal (inclusionwise) subsets of S: $\{1,4\}, \{2,3\}$ which has the same cardinality.
So is this example a matroid? **NO!**.
$A = \{1,2,3\}$, then both $\{1\}, \{2,3\}$ are the maximal subsets of A that are in I but they have different cardinality.

Definition 8: nomenclature of matroids

- Elements of I are called independent sets.
- Minimal dependent sets are called circuit (in the forest sense, cycle are circuits).
- If $M = (S, I)$ satisfies $(M1), (M2)$, it's called an independence system.
- The rank of A: $r(A) := \max\{|B|\colon B \subseteq A, B \in I\}$
- The basis of M = the basis of S.
- $r(S)$ is the rank of M (matroid or independent system).
- $\rho(A) := \min\{|B|\colon B \text{ is a basis of } A\}$. Note:

$$M \text{ is a matroid} \iff \rho(A) = r(A), \forall A \subseteq S$$

2.2 Matroid 2

Maximum Weight independent Set (for independent systems):
Given $M = (S, I)$ independence system, $c_e \in \mathbb{R}_+$ (I can always delete the ones with $c_e < 0$, so assume $c_e \geqslant 0$), for all $e \in S$, find $A \in I$ maximizing $c(A) := \sum_{e \in A} C_e$.

$F \leftarrow \emptyset$
while $\exists e : F \cup \{e\} \in I$ and $c_e > 0$ **do**:
  Choose such e with largest c_e;
  $F \leftarrow F \cup \{e\}$
end while
  return F

Theorem 9: Rado, Edmonds

Let M be a matroid, $c \in \mathbb{R}_+^S$. Then greedy algorithm above finds Maximum Weight Independent Set.

Proof. Later □

Theorem 10: Rado, Edmonds

Let $M = (S, I)$ be an independence system. Then greedy finds an optimal independent set $\forall c \in \mathbb{R}_+^S$ if and only if M is a matroid.

Proof.

- ($\Longleftarrow$) By Theorem 9 above.
- ($\Longrightarrow$) Suppose M is not a matroid. Let $A \subseteq S$, A_1 and A_2 be bases of A with $|A_1| < |A_2|$. Let
$$c_e = \begin{cases} v_1, & \forall e \in A_1 \\ v_2, & \forall e \in A_2 \setminus A_1 \\ 0, & \forall e \notin A_1 \cup A_2 \end{cases}$$
Choose $v_1 > 0$ and $v_1 > v_2 > \frac{|A_1|}{|A_2|} v_1$. Then since all other elements have cost zero, the greedy algorithm only considers the elements in $A_1 \cup A_2$. Since $v_1 > v_2$, the algorithm will select A_1 first, since A_1 is a basis of A, the algorithm can't add more elements to it, so it stops and output A_1. Then A_2 has cost $v_1|A_1 \cap A_2| + v_2|A_2 \setminus A_1| \geqslant v_2|A_2| > v_1|A_1|$. So the greedy algorithm does not ouput an optimal solution when all $c \in \mathbb{R}_+^S$, contradiction.

□

Theorem 11: Jenkyns 176

Let (S, I) be an independent system. Let $gr_{S,I}$ be the total weight of the independent set formed by the greedy algorithm and $opt_{S,I}$ be the optimal solution weight. Then

$$gr_{S,I} \geqslant q(S, I) opt_{S,I}$$

where $q(S, I) = \min_{A \subseteq S, r(A) \neq 0} \frac{\rho(A)}{r(A)}$ (rank quotient).

Proof. Let $S = \{e_1, \ldots, e_n\} : c_{e_1} \geqslant \ldots \geqslant c_{e_n}$. Let $S_j := \{e_1, \ldots, e_j\}$ and $S_0; = \emptyset$. Let $G \in I$ be solution obtained by greedy, $\sigma \in I$ be the optimal solution and $G_j = G \cap S_j$; $\sigma_j = \sigma \cap S_j$.

$$c(G) = \sum_{j \in G} c_j = \sum_{j=1}^{n} c_{e_j}(|G_j| - |G_{j-1}|) = \sum_{j=1}^{n} |G_j| \underbrace{(c_{e_j} - c_{e_{j+1}})}_{\Delta_j \geqslant 0}$$

note that if $e_j \in G$, then $|G_j| - |G_{j-1}| = 1$, otherwise, $|G_j| - |G_{j-1}| = 0$ and $c_{e_{n+1}} := 0$.
Greedy computes a maximum independent subset of S_j implies G_j is a basis of S_j implies

$$\begin{aligned}
c(G) &= \sum_{j=1}^{n} |G_j| \Delta_j \\
&\geqslant \sum_{j=1}^{n} \rho(S_j) \Delta_j \\
&\geqslant \sum_{j=1}^{n} q(S, I) r(S_j) \Delta_j \\
&\geqslant \sum_{j=1}^{n} q(S, I) |\sigma_j| \Delta_j \\
&= q(S, I) \sum_{j=1}^{n} |\sigma_j| (c_{e_j} - c_{e_{j+1}}) \\
&= q(S, I) \sum_{j=1}^{n} c_{e_j} (|\sigma_j| - |\sigma_{j-1}|) \\
&= q(S, I) \sum_{j \in \sigma} c_j \\
&= q(S, I) c(\sigma)
\end{aligned}$$

□

Hence, by Jenkyn's results, we have if M is a matroid, greedy gets an optimal solution. And **Theorem 9 is proved by it**.
How fast is Greedy? Hence a total $O(|S|)$ times executed.

$F \leftarrow \emptyset$ $O(1)$
while $\underbrace{\exists e : F \cup \{e\} \in I}_{\text{can be checked in time } Poly(|S|)?}$ and $c_e > 0$ **do**:
  Choose such e with largest c_e; $O(|S|)$
  $F \leftarrow F \cup \{e\}$ $O(1)$
end while
  return F $O(1)$

2.3 Matroid 3

Theorem 12

Let $M = (S, I)$ independent system. Then $(M3) \iff (M3') : \forall X, Y \in I, |X| > |Y|, \exists x \in X \setminus Y : Y \cup \{x\} \in I$.

Proof.

- $(M3') \implies (M3)$ trivial.
- $(M3) \implies (M3')$. Let $X, Y \in I$ and $|X| > |Y|$. Consider $A = X \cup Y$. Then Y is not a basis of A because by $(M3)$, and $|X| > |Y|$, we have $|Y| < r(A)$. Then there exists $x \in A \setminus Y = X \setminus Y : Y \cup \{x\} \in I$.

□

Example 13

Let $G = (V, E), W \subseteq V$ a stable set. Let $k_v \in \mathbb{Z}_+, \forall v \in W, S = E, I = \{F \subseteq E : |\delta(v) \cap F| \leqslant k_v, \forall v \in W\}$. Clearly $(M1), (M2)$ hold.
$(M3')$ Let $X, Y \subseteq E$, $X, Y \in I$, $|X| > |Y|$. Let $W_Y = \{v \in W : |\delta(v) \cap Y| = k_v\}$. Also, $2|X| = \sum_{v \in V} |X \cap \delta(v)|$. then

$$2|X| = \sum_{v \in W_Y} \underbrace{|X \cap \delta(v)|}_{\leqslant k_v} + \sum_{v \in W \setminus W_Y} |X \cap \delta(v)| + \sum_{v \in V \setminus W} |X \cap \delta(v)|$$
$$2|Y| = \sum_{v \in W_Y} \underbrace{|Y \cap \delta(v)|}_{= k_v} + \sum_{v \in W \setminus W_Y} |Y \cap \delta(v)| + \sum_{v \in V \setminus W} |Y \cap \delta(v)|$$

Since $|X| > |Y|$, there exists $x \in X \setminus Y$: $x \in \delta(v)$ only for some $v \notin W_Y$. Otherwise, all $x \in X$ are either in Y or incident to W_Y, then $|X|$ is the number of edges in X incident to W_Y and the rest. While the rest part of X are all in Y but not incident to W_Y which are in the set of edges in Y but not incident to W_Y, and the number of edges in X incident to W_Y is less than or equal to number of edges in Y incident to W_Y. Mathematically, say K_X is subset of X such that $x \in K_X \iff x \in \delta(v)$ for some $v \in W_Y$. K_Y is the subset such that $y \in K_Y \iff y \in \delta(v)$ for some $v \in W_Y$. And $|K_X| \leqslant |K_Y|$ by the definition of W_Y. Then, $X \setminus K_X \subseteq Y \setminus K_Y$. Hence, $|X| \leqslant |Y|$, contraction. Then $Y \cup \{x\}$ satisfies the condition.

2.3.1 Circuit characterization

Theorem 14: Circuit

If instead of describing I, you are given the set of circuits (min. dependent set) ($\mathcal{C}$) of M, then $A \in I \iff \nexists c \in \mathcal{C} : c \subseteq A$.

Proof.

- $\Longrightarrow$: by $(M2)$, any subset of A should be in I, so it has no subset in $\mathcal{C}$.
- $\Longleftarrow$: Suppose A is not in I, then its dependent, keep deleting elements from A till it's in $\mathcal{C}$, then we have a subset of A which is in $\mathcal{C}$, contradiction.

$\square$

Example 15

$S = \{1, 2, 3, 4\}, \mathcal{C} = \{\{4\}, \{1, 2, 3\}\}, I = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}\}$.

Q: When is $\mathcal{C} \subseteq 2^S$ the set of circuits of a matroid?

Theorem 16

Let $M = (S, I)$ be a matroid. Then $\forall A \in I, \forall e \in S$, $A \cup \{e\}$ contains at most 1 circuit.

Proof. Let A be smallest set so that

- $A \in I$
- $\exists e : A \cup \{e\}$ has two distinct circuits C_1, C_2.

Note $e \in C_1 \cap C_2$, otherwise, A has a circuit then it can't be in I.
By the choice of A, we have $A \cup \{e\} = C_1 \cup C_2$ (otherwise there exists $u \in A \setminus (C_1 \cup C_2)$, then $A \setminus \{u\}$ is a smaller set satisfying the properties above).
Since $C_1 \nsubseteq C_2, C_2 \nsubseteq C_1$ (if $C_1 \subset C_2$, then C_2 is not a circuit), let $e_1 \in C_1 \setminus C_2$ and $e_2 \in C_2 \setminus C_1$.
Consider $A' = (C_1 \cup C_2) \setminus \{e_1, e_2\}$, if A' has a circuit C, then

- $C \neq C_1$, $C \neq C_2$ because $C \cap \{e_1, e_2\} = \emptyset$.
- Since $e_1 \notin C_2$, $C_2 \subseteq \{A \setminus e_1\} \cup \{e\}$, similarly, C is also a subset of it.
- Then $A \setminus \{e_1, e_2\}$ will be a set satisfying the properties, contradicts to the minimality of A, so A' has no circuit, so $A' \in L$.

so A, A' are bases of $C_1 \cup C_2$, with $|A'| < |A|$ which contradicts to M being a matroid. $\square$

Theorem 17

Let $\mathcal{C} \subseteq 2^S$. Then $\mathcal{C}$ is the set of circuits of a matroid iff

(C1) $\emptyset \notin \mathcal{C}$

(C2) If $C_1, C_2 \in \mathcal{C}, C_1 \subseteq C_2$, then $C_1 = C_2$.

(C3) If $C_1, C_2 \in \mathcal{C}, C_1 \neq C_2$ and $e \in C_1 \cap C_2$, then there exists $C \in \mathcal{C}$ with $C \subseteq (C_1 \cup C_2) \setminus \{e\}$.

Proof.

- ($\implies$): $(C_1), (C_2)$ are trivial to prove. Suppose $(C3)$ not true, then $A := (C_1 \cup C_2) \setminus \{e\} \in I$ by $\nexists c \in \mathcal{C}$ such that $C \subseteq A$. This implies $A \cup \{e\}$ has two distinct circuits, contradicts to the previous theorem.

- Define $I = \{A \subseteq S : \nexists C \in \mathcal{C} \text{ with } C \subseteq A\}$. Let $M = (S, I)$, then $(M1)$, $(M2)$ clearly hold.
  Suppose $(M3)$ is false, let A_1, A_2 be the bases of $A \subseteq S$ with $|A_1| < |A_2|$, choose A_1, A_2 with largest $|A_1 \cap A_2|$. Let $e \in A_1 \setminus A_2$ (it exists because $A_1 \nsubseteq A_2$) and $A_2 \cup \{e\}$ contains a circuit C. If $A_2 \cup \{e\}$ contains $C' \neq C$ (note $e \in C \cap C'$), then $(C3) \implies A_2$ contains a circuit, but $A_2 \in I$. Hence C is a unique circuit in $A_2 \cup \{e\}$. Let $f \in C \setminus A_1 \implies \underbrace{(A_2 \cup \{e\}) \setminus \{f\}}_{A_3} \in I$, but $|A_3 \cap A_1| > |A_2 \cap A_1|, |A_3| = |A_2| > |A_1|$, contradiction. Note: we can make A_3 a basis by adding elements, but the inequalities above still hold.

$\square$

2.4 Matroid 4

Theorem 18: Bases characterization

Instead of giving I, we are given $\mathbb{B}$, the set of bases of M, then $A \in I \iff A \subseteq B$, for some $B \in \mathbb{B}$.

Theorem 19

Let $\mathbb{B} \subseteq 2^S$. $\mathbb{B}$ is the set of bases of a matroid (S, I) if and only if

(B1) $\mathbb{B} \neq \emptyset$

(B2) $\forall B_1, B_2 \in \mathbb{B}$, $x \in B_1 \setminus B_2$, there exists $y \in B_2 \setminus B_1$ such that $(B_1 \setminus \{x\}) \cup \{y\} \in \mathbb{B}$.

Theorem 20

Let $\mathbb{B} \subseteq 2^S$. $\mathbb{B}$ is the set of bases of a matroid (S, I) if and only if

(B1) $\mathbb{B} \neq \emptyset$

(B2) $\forall B_1, B_2 \in \mathbb{B}$, $y \in B_2 \setminus B_1$, there exists $x \in B_1 \setminus B_2$ such that $(B_1 \setminus \{x\}) \cup \{y\} \in \mathbb{B}$.

2.5 Polymatroids

Let $M = (S, I)$ be a matroid, $c \in \mathbb{R}^S_+$. Let $x \in \mathbb{R}^S$ be decision variables.

$$(P_M) \qquad \begin{aligned} &\max c^T x \\ &s.t.\ x(A) \leqslant r(A), \forall A \subseteq S \\ &x \geqslant 0 \end{aligned}$$

Note: If $J \in I$, then x^J (incidence vector) is feasible for (P_M).

Theorem 21

Let $M = (S, I)$ be a matroid, and let G be the solution returned by the greedy algorithm. Then x^G is optimal for (P_M).

Definition 22

A function $f : 2^S \mapsto \mathbb{R}$ is called submodular if $\forall A, B \subseteq S$,

$$f(A) + f(B) \geqslant f(A \cap B) + f(A \cup B)$$

Proposition 23

Let $M = (S, I)$. Then $r(A)$ is submodular.

Proof. Let $A, B \subseteq S$. Let $J_\cap$ be a basis of $A \cap B$ (i.e. $|J_\cap| = r(A \cap B)$). Extend $J_\cap$ to a basis J_B of B (i.e. $|J_B| = r(B)$) (by keep adding elements of B to $J_\cap$ until we get a maximal independent set contained in B). Similarly, extend J_B to a basis $J_\cup$ of $A \cup B$ (i.e. $|J_\cup| = r(A \cup B)$). Let $J' = J_\cup \setminus (J_B \setminus J_\cap)$

- Since $J' \subseteq J_\cup$, we have $J' \in I$.
- Suppose there exists $v \in J' \setminus A$, then $v \in J_\cup \setminus A$ and $v \notin J_B \setminus J_\cap$. Since $v \notin A$, we have $v \notin J_\cap$. so $v \notin J_B$, and $v \in B$. Since $J_\cup$ is a basis, $J_B \cup \{v\} \in I$, then J_B is not a basis of B, contradiction. So $J' \subseteq A$.

Thus:

$$r(A) + r(B) \geqslant |J'| + |J_B| = |J_\cup - (|J_B| - |J_\cap|) + |J_B| = |J_\cup| + |J_\cap| = r(A \cup B) + r(A \cap B)$$

□

Definition 24

Let $f : 2^S \mapsto \mathbb{R}_+$ be submodular, then

$$\left\{x \in \mathbb{R}^S : \substack{x(A) \leqslant f(A) \\ x \geqslant 0}, \forall A \subseteq S\right\}$$

is called a Polymatroid.

Note: May assume $f(\emptyset) = 0$, f is monotone (i.e. $X \subseteq Y \subseteq S \impliedby f(X) \leqslant f(Y)$). Consider (where f monotone and $f(\emptyset) = 0$)

$$(P_f) \quad \begin{aligned} &\max c^T x \\ &s.t.\ x(A) \leqslant f(A), \forall A \subseteq S \\ &x \geqslant 0 \end{aligned}$$

$$
\begin{aligned}
& \min \sum_{A \subseteq S} f(A) y_A \\
(D_f) \quad & s.t. \sum_{A: e \in A} y_A \geqslant c_e, \forall e \in S \\
& y \geqslant 0
\end{aligned}
$$

Primal Greedy

$S = \{e_1, \ldots, e_n\}$, $c_{e_1} \geqslant \ldots \geqslant c_{e_k} \geqslant 0 \geqslant c_{e_{k+1}} \geqslant \ldots \geqslant c_{e_n}$. $S_j = \{e_1, \ldots, e_j\}$ and

$$
x_{e_j} = \begin{cases} f(S_j) - f(S_{j-1}), & \forall j = 1, \ldots, k \\ 0, & \forall j > k \end{cases}
$$

If $f(S_j) = r(S_j)$, for $M = (S, I)$ matroid. Let G be a greedy solution, $G_j := G \cap S_j$. If G_{j-1} is a basis of S_{j-1}, then when $r(S_j) = r(S_{j-1})$, we have $x_{e_j} = 0$, so $e_j \notin G_j$. Then $G_j = G_{j-1}$, so G_{j-1} is also a basis of S_j. When $r(S_j) > r(S_{j-1})$, $x_{e_j} = 1$, so $e_j \in G \implies G_j = G_{j-1} \cup \{e_j\} \implies G_j$ is a basis of S_j.

Dual Greedy

$$
\begin{aligned}
& y_{S_j} = c_{e_j} - c_{e_{j+1}}, \forall j = 1, \ldots, k-1 \\
& y_{s_k} = c_{e_k} \\
& y_A = 0, \text{ for all other } A
\end{aligned}
$$

We can show that x, y above are optimal solutions by Complementary Slackness conditions.

Corollary 25

Let $M = (S, I)$, $c \in \mathbb{R}^S$, $J \in I$. Then J is an inclusionwise minimal, max weight independent set if and only if

(a) $e \in J \implies c_e > 0$

(b) $e \notin J, J \cup \{e\} \in I \implies c_e \leqslant 0$

(c) $e \notin J, f \in J, (J \cup \{e\}) \setminus \{f\} \in I \implies c_e \leqslant c_f$.

Proof.

- $\implies$: trivial
- $\impliedby$: Consider (P_r), where r is the rank function of M which is monotone, submodular and $r(\emptyset) = 0$. (note J is independent, so J is feasible for P_r). Let y be the solution from greedy. Let x^J be the characteristic vector of J. Then

$$
\sum_{A: e_j \in A} y_A = c_{e_j}, \forall j \leqslant k
$$

and

$$a) \implies x^J e_j = 0, \forall j > k$$

Thus for all $j \in \{1, \ldots, n\}$, we have $x^J_{e_j} = 0$ OR $\sum_{A: e_j \in A} y_A = c_{e_j}$.
Pick $y_A > 0$. By construction, $A = S_j$ for $j \leqslant k$. Note $x^J(S_j) = |J \cap S_j| = |J_j|$. Suppose $|J_j| < r(S_j)$, then J_j is not a basis of S_j, but it's an independent set so there exists $e \in S_j \setminus J$ such that $J_j \cup \{e\} \in I$.

Case 1 $J \cup \{e\} \in I$, then $b \implies c_e \leqslant 0$, but $e \in S_j \implies c_e > 0$, contradiction.

Case 2 $J \cup \{e\} \notin I$. Extend $J_j \cup \{e\}$ to a basis J' of $J \cup \{e\}$. Note J is a basis of $J \cup \{e\}$. Hence, $|J'| = |J|$ by both being basis of $J \cup \{e\}$.
Then there exists $f \in J \setminus S_j$ such that $J' = (J \cup \{e\}) \setminus \{f\} \in I$. This f exists because $e \in J' \setminus J$, so there is $f \in J \setminus J'$, and $J_j \subseteq J \cap J'$, so $f \notin J_j$, which implies $f \notin S_j$. Then by $c)$, $c_e \leqslant c_f$.
By $y_{S_j} = c_{e_j} - c_{e_{j+1}} > 0 \implies c_{e_j} > c_{e_{j+1}}$ and $f \notin S_j \implies c_{e_{j+1}} \geqslant c_f \geqslant c_e$, we have $c_{e_j} > c_{e_{j+1}} \geqslant c_f \geqslant c_e$, but $e \in S_j$, so $c_e \leqslant c_{e_j}$, contradiction.

Hence, $x^J(S_j) = r(S_j)$.

Hence, Complementary Slackness conditions hold, so x^J is optimal for (P_r) which implies J is a maximal weight independent set. And $a)$ implies the inclusionwise minimality. □

2.6 Matroid Construction

Let $M = (S, \mathcal{I})$ be a matroid.

1. Deletion: Let $B \subseteq S$, $M \setminus B := (S', \mathcal{I}')$ is a matroid, where $S' := S \setminus B$, $\mathcal{I}' := \{A \subseteq S \setminus B : A \in \mathcal{I}\}$.

2. Truncation: Let $k \in \mathbb{Z}_+$, define $\mathcal{I}' := \{A \in \mathcal{I} : |A| \leqslant k\}$. Then $M' = (S, \mathcal{I}')$ is a matroid.

3. Disjoint Union: Let $M_i = (S_i, \mathcal{I}_i), \forall i \in \{1, \ldots, k\}$ be matroids, with $S_i \cap S_j = \emptyset, \forall i \neq j$. Then $M_1 \oplus \ldots \oplus M_k = (S, \mathcal{I})$, with $S = \cup_{i=1}^k S_i$. And $\mathcal{I} = \{A \subseteq S : A \cap S_i \in \mathcal{I}_i, \forall i = 1, \ldots, k\}$ is a matroid.

Proof.

(M1) hold

(M2) hold

(M3) Let B be a basis of $A \subseteq S$. Let $B_i = B \cap S_i, \forall i$. Then $B_i \in \mathcal{I}$, but also, it is a basis of $A \cap S_i$. (otherwise, we can add $\alpha \in A \cap S_i$ to B_i, hence to B, then B is not a basis of A). This implies $|B| = \sum_{i=1}^k |B_i| = \sum_{i=1}^k r_i(A \cap S_i)$, thus all basis of A have the same size.

□

Example: Partition Matroid: Let $S = S_1 \dot\cup S_2 \dots \dot\cup S_k$, $b_1, \dots, b_k \in Z_+$. $M = (S, \mathcal{I})$ where $I = \{A \subseteq S : |A \cap S_i| \leqslant b_i, \forall i = 1, \dots, k\}$. Then $M_i = (S_i, \mathcal{I}_i), \mathcal{I}_i = \{J \subseteq S_i : |J| \leqslant b_i\}$ is the uniform matroid. And $M = M_1 \oplus \dots \oplus M_k$.

4. Contraction: Let $B \subseteq S$, let J be a basis of B. Then $M/B = (S', \mathcal{I}')$ where $S' = S \setminus B$, $\mathcal{I}' = \{A \subseteq S' : A \cup J \in \mathcal{I}\}$.

Proposition 26

If M is forest matroid of $G = (V, E)$, $B \subseteq E$, then M/B is a forest matroid of G/B (contraction in graph theory).

Theorem 27

M/B is a matroid independent from choice of J, and its rank fcn is $r_{M/B}(A) = r_M(A \cup B) - r_M(B)$.

Proof. $(M1), (M2)$ clearly hold. $(M3)$: Let $A \subseteq S' = S \setminus B$, let J' be an M/B basis of $A \implies J \cup J' \in \mathcal{I}$.

Claim. $J \cup J'$ is an M-basis of $A \cup B$.

Proof. Suppose not, then there is $e \in A \cup B \setminus J \cup J'$ and $J \cup J' \cup \{e\} \in \mathcal{I}$. If $e \in B$, then $J \cup \{e\} \in \mathcal{I}$ and it's a subset of B, contradicts to J being a basis of B. If $e \notin B$ then $e \in A \setminus B$, then $(J' \cup \{e\}) \cup J \in \mathcal{I} \implies J' \cup \{e\} \in \mathcal{I}'$, contradicts to J' being a basis of A. $\square$

By the claim, $|J \cup J'| = r_M(A \cup B) \implies |J| + |J'| = r_M(A \cup B) \implies |J'| = r_M(A \cup B) - |J| = r_M(A \cup B) - r_M(B)$. Thus, $A \in \mathcal{I}'$ if and only if $|A| = r_{M/B}(A) = r_M(A \cup B) - r_M(B)$ which doesn't depend on J. $\square$

5. Duality: $M^* = (S, \mathcal{I}^*), \mathcal{I}^* = \{A \subseteq S : S \setminus A$ has a basis of $M\} = \{A \subseteq S : r_M(S \setminus A) = r_M(S)\}$. Note here the basis of M means the basis of S.
Example: $M = U_n^r$, $A \subseteq S = \{1, \dots, n\}, A \in \mathcal{I}^* \iff |A| \leqslant n - r \implies M^* = U_n^{n-r}$.

Theorem 28

M^* is a matroid with rank function $r^*(A) = |A| + r_M(S \setminus A) - r_M(S)$.

Proof. Clearly $(M1), (M2)$ hold. For $(M3)$, let $A \subseteq S$, let J^* an M^*-basis of A. Let J be an M-basis of $S \setminus A$. Extend J to an M-basis J' of $S \setminus J^*$. By definition, we know J' is an M-basis of S.

Claim. $A \setminus J^* \subseteq J'$

Proof. Suppose $e \in (A \setminus J^*) \setminus J' \implies J' \subseteq S \setminus (J^* \cup \{e\})$, and since J' is an M-basis of S, $J^* \cup \{e\} \in I^*$, contradiction. $\square$

Then

$$|J'| = |A \setminus J^*| + |J| = |A| - |J^*| + |J| \iff |J^*| = |A| - |J'| + |J| = |A| - r_M(S) + r_M(S \setminus A)$$

$\square$

3 Matchings

3.1 Matchings 1

Definition 29

Given a graph $G = (V, E)$, a subset $M \subseteq E$ is a matching if $|\delta(v) \cap M| \leqslant 1, \forall v \in V$; i.e., every vertex incident to at most one edge in M.
Given a matching M, a vertex v is called M-covered if $|\delta(v) \cap M| = 1$, and it's called M-exposed otherwise.

Note: There are $2|M|$ M-covered vertices and $|V| - 2|M|$ M-exposed vertices.

- A matching is perfect if there are no M-exposed vertices.
- The size of the largest cardinality matching in G will be denoted as $\nu(G)$. (M is a perfect matching if and only $\nu(G) = \frac{|V(G)|}{2}$.)
- Given $G = (V, E)$, and a matching M, a path $P = (v_1, \ldots, v_k)$ is called M-alternating if $\{v_{i-1}, v_i\} \in M \iff \{v_i, v_{i+1}\} \notin M, \forall i = 2, \ldots, k-1$.
- An M-alternating path is called M-augmenting if v_1, v_k are exposed.
- Given $F_1, F_2 \subseteq E$, the symmetric difference between F_1, F_2 is defined as

$$F_1 \triangle F_2 = \{e \in E : e \text{ is in exactly one of } F_1, F_2\}$$

Theorem 30

Let M be a matching of $G = (V, E)$. Then M is a max cardinality matching if and only if there does not exist an M-augmenting path.

Proof.

- ($\implies$) Suppose there exists an M-augmenting path $P = \{v_0, \ldots, v_k\}$. Let $e_i = \{v_{i-1}, v_i\}$, $\forall i = 1, \ldots, k$. Let $M' = M \triangle E(P)$. Note since P is M-augmenting, we know v_0, v_k are M-exposed, so $e_1, e_k \notin M$, so k is odd. That is $|M'| = |(E(P) \setminus M) \cup (M \setminus E(P))| = |M| - |(E(P) \cap M)| + |(E(P) \cap M)| + 1 = |M| + 1$. Suppose M' is not a matching. Then there are two edges in M' incident to one vertex. If both edges are in M, then M is not a matching, contradiction. Hence, at lease one of them is in $M' \setminus M$, call it e. Then e is in $E(P) \setminus M$. If we have $v_i v_{i+1} e v_{i+2} v_{i+3}$. Then $v_i v_{i+1}$ and $v_{i+2} v_{i+3}$ are not in M' by they are in both M and $E(P)$, so v_{i+1}, v_{i+2} are not M'-exposed, contradiction. If one end of e is v_0 (v_k), then if v_1 incident to another egde in e', we know $e' \neq v_1 v_2$ by $v_1 v_2 \in M \implies v_1 v_2 \notin M'$, so $e' \notin E(P)$, so $e' \in M$, then v_1 incidence to e' and $v_1 v_2$ in M, contradiction. Hence, v_0 is incidence to another e' in M', then $e' \notin E(P) \implies e' \in M$, but v_0 is M-exposed, contradiction. Hence, M' is a matching, and $|M'| > |M|$, contradiction, so not such P exists.

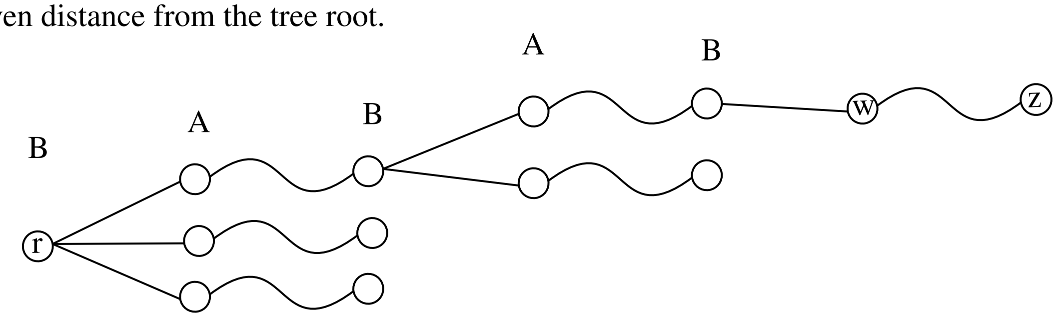
- ($\Longleftarrow$) Suppose M' is a matching of G with $|M'| > |M|$. Consider $G' = (V, M \triangle M')$. Note $|\delta_{G'}(v)| \leqslant 2, \forall v \in V$, because if $|\delta_{G'}(v)| = 3$ for some $v \in V$, then there are three edges incident to it, so there are at least two edges in the same matching incident to it, contradiction. Also $|\delta_{G'}(v)| \leqslant 2 \implies G'$ is a (edge) disjoint union of paths and cycles, and all of them are alternating (w.r.t. M and M'). Also note if C is a cycle in G', $|E(C) \cap M| = |E(C) \cap M'|$, otherwise C is an odd cycle and there will be a vertex incident to two edges in M or M', contradiction. Hence, there exists a path P with $|E(P) \cap M'| > |E(P) \cap M|$, then P is the desired M-augmenting path in G, contradiction.

$\square$

- Q: Does there exist a path from a vertex u to a vertex v?
  A: Use Breadth First Search.

- Q: Does there exist an M-alternating path from an M-exposed vertex u to an M-exposed vertex v?
  A: Similar, keep the path you are looking for alternating. Instead of constructing a Breadth First Search Tree, we construct an "alternating" trees. It can keep track of nodes at odd/even distance from the tree root.

Tentative Algorithm:
Input: $G = (V, E)$, M is a matching, $r \in V$ is M-exposed. $T \leftarrow (\{r\}, \emptyset)$, $A(T) \leftarrow \emptyset$, $B(T) \leftarrow \{r\}$, where A represents the node at odd distance from the tree root, and B represents the node at even distance from the tree root.



In the tree, we use tilde lines to represent the edges in M and straight lines otherwise. Also, we can see that each path from the root r to a node in T is an M-alternating path in G.

Case 1: If we can find $vw \in E$: $v \in B(T), w \notin V(T)$, and w is M-covered. We can extend T using vw.
Let $z \in V : wz \in M$, since every vertex in T, v is either incident to another edge in M or M-exposed, $z \notin V(T)$,. Then update $V(T) \leftarrow V(T) \cup \{w, z\}$, $B(T) \leftarrow B(T) \cup \{z\}$, $A(T) \leftarrow A(T) \cup \{w\}$, $E(T) \leftarrow E(T) \cup \{vw, wz\}$.

Case 2: If we find $vw \in E : v \in B(T), w \notin V(T)$ and w is M-exposed, then we find an M-augmenting path from r to w, which is $P' = P + vw$, where P is the M-alternating path from r to v in T, $M \leftarrow M \triangle P'$.

Hence, the tentative algorithm can be written as

Algorithm 5 Tentative Algorithm for Matchings

$G = (V, E)$, M is a matching, $r \in V$ and it's M-exposed. $T \leftarrow (\{r\}, \emptyset)$, $A(T) \leftarrow \emptyset$, $B(T) \leftarrow \{r\}$ (initialized T with r).
while $\exists vw \in E : v \in B(T), w \notin V(T)$ **do**:
  if w is M-covered **then**
    Use vw to extend T
  else
    Use vw to augment M;
    if $\exists$ M-exposed vertex $r \in V$ **then**
      Initialize T with r
    else
      Stop
    end if
  end if
end while
  return M

This does not always work (e.g. G is not connected).

3.2 Matching 2

Definition 31

A graph is <u>bipartite</u> if there exists a partition (A, B) of V such that $\forall e \in E, |e \cap A| = |e \cap B| = 1$.

Theorem 32: Hall's Theorem

Let $G = (V, E)$ be bipartite, with bipartition $V = A \dot\cup B$. Then there exists a matching covering A if and only if $|N(X)| \geqslant |X|, \forall X \subseteq A$, where $N(X) := \{v \in V \setminus X : \exists u \in X \text{ with } \{u, v\} \in E\}$.

Proof.

- ($\implies$) If there exists $X \subseteq A : |X| > |N(X)|$, then since vertices only matched to vertices in $N(X)$, no matching can cover all vertices in X (there is a vertex in X having no neighbors).

- ($\impliedby$) By induction, cases $|A| = 0, |A| = 1$ are trivial.
  If $|N(X)| - |X| > 0, \forall X \subset A, X \neq \emptyset$, pick $uv \in E$ with $u \in A$, $v \in B$, and consider $G' = G \setminus \{u, v\}$, bipartite with bipartition $A' = A \setminus \{u\}$, $B' = B \setminus \{v\}$. Now, $\forall X \subseteq A'$, $|N_{G'}(X)| \geqslant |N_G(X)| - 1 \implies |N_{G'}(X)| - |X| \geqslant 0, \forall X \subseteq A'$. By induction, there exist a matching M' covering A', then $M' \cup \{u, v\}$ covers A.

If $|N(X)| = |X|$, for some $X \subset A, X \neq \emptyset$. By induction, there exist a matching M_1 in $G[X \cup N(X)]$ covering X. Now consider $G' = G[(A \backslash X) \cup (B \backslash N(X))]$. Note $\forall Y \subseteq A \backslash X$,

$$|N_{G'}(Y)| = |N_G(Y) \setminus N_G(X)| = |N_G(X \cup Y)| - \underbrace{|X|}_{=|N_G(X)|} \geqslant |X \cup Y| - |X| = |Y|$$

Hence, there exists a matching in G' covering $A \backslash X$, combine it with M_1, there is a matching covering A.

$\square$

Corollary 33

Let $G = (V, E)$ be bipartite with bipartition $V = A \dot{\cup} B$. Then G has a perfect matching if and only if $|A| = |B|$ and $|X| \leqslant |N(X)|, \forall X \subseteq A$.

Algorithm 6 Algorithm for Perfect Matchings of bipartite graphs

---

```
G = (V, E) be bipartite, initialize T with r.
while ∃vw ∈ E : v ∈ B(T), w ∉ V(T) do:
    if w is M-covered then
        Use vw to extend T
    else
        Use vw to augment M;
        if ∃ M-exposed vertex r ∈ V then
            Initialize T with r
        else
            Stop, output perfect matching M
        end if
    end if
end while
Output No Perfect Matchings exists. (*)
```

If algorithm reaches (*), then G has no perfect matching.

Proof. If the algorithm reaches (*), then

- $N(B(T)) = A(T)$. First, $A(T) \subseteq N(B(T))$, and if there exist a vertex $u \in N(B(T)) \backslash A(T)$ which is a neighbor of $v \in B(T)$, then $u \notin B(T)$, because otherwise, both u, v are at even distance from the root, and by G being bipartite, that means both u, v are in the same partition of G, and they are incident, contradiction.

- $|B(T)| > |A(T)|$. Suppose the tree has a leaf in $A(T)$, then by our algorithm, if it's M-exposed, we augment M, otherwise, we extend T, so all leaves of T are in $B(T)$. That is, for every vertex in $A(T)$, it has a neighbor in $B(T)$ in the tree with one larger height from the root, and since $r \in B(T)$, we have $|B(T)| > |A(T)|$.

- By what's above, we know $|B(T)| > |N(B(T))|$, by the Corollary above, G has no perfect matching.

□

Definition 34

$U \subseteq V$ is a vertex cover if $\forall e \in E, |e \cap U| \geqslant 1$. We let $\tau(G)$ be the size of the smallest cardinality vertex cover. Fact: $\nu(G) \leqslant \tau(G)$. Otherwise, consider the max cardinality matching, you need at least $|M|$ vertice to cover the M-covered vertices because for each edge in M, you need one of the ends in the vertex cover.

Theorem 35: König's Theorem

Let G be bipartite, then $\nu(G) = \tau(G)$.

3.3 Matching 3

Recall $\nu(G) \leqslant \tau(G)$ and equality holds for bipartite graph. Suppose $A \subseteq V$, let $H_1, \ldots, H_k$ be odd connected components of $G \setminus A$.
Q: How many M-exposed vertices can there be?
If H_i has no M-exposed vertices, then there exists at least on edge in M from H_i to A (becasue there are odd number of vertices in H_i). But there are at most $|A|$ such edges, implies there are at least $k - |A|$ M-exposed vertices for all matching M.
Recall: there are $|V| - 2|M|$ M-exposed vertices in any matching, which implies $|V| - 2|M| \geqslant k - |A|, \forall M$. It is equivalent to $|M| \leqslant \frac{1}{2}(|V| - k + |A|)$. Then, let $k = oc(G \setminus A)$ (number of odd components of $G \setminus A$),

$$\nu(G) \leqslant \frac{1}{2}(|V| - oc(G \setminus A) + |A|), \forall A \subseteq V$$

We also note that if A is a vertex cover, then $G \backslash A$ is a graph with no edges, so $oc(G \backslash A) = |V| - |A|$, then the bound above becomes $|A|$.

Theorem 36: Tutte-Berge Formula

Let $G = (V, E)$ be a graph. Then

$$\max\{|M| : M \text{ is a matching}\} = \frac{1}{2} \min\{|V| - oc(G \setminus A) + |A| : A \subseteq V\}$$

Theorem 37: Tutte's Matching Theorem

G has a perfect matching $\iff oc(G \setminus A) \leqslant |A|, \forall A \subseteq V$.

Proof. If $oc(G) > 0$, then G has no perfect matching and $A = \emptyset$ violates $oc(G \setminus A) \leqslant |A|$.

If $oc(G) = 0$, then

$$\begin{aligned}
&G \text{ has a perfect matching}\\
\iff &\nu(G) = \frac{n}{2}\\
\iff &n = \min\{n - oc(G \setminus A) + |A|\colon A \subseteq V\}\\
\iff &\min\{|A| - oc(G \setminus A) : A \subseteq V\} = 0
\end{aligned}$$

But for $A = \emptyset$, $|A| - oc(G \setminus A) = 0$, so 0 can be obtained, that is,

$$\min\{|A| - oc(G \setminus A) : A \subseteq V\} = 0 \iff oc(G \setminus A) \leqslant |A|, \forall A \subseteq V$$

□

So Tutte's Matching Theorem is proved by using Tutte-Berge Formula, which is what we want to prove now. Before that, we say $u \in V$ is essential if u is M-covered in <u>EVERY</u> maxmimum cardinality matching M; otherwise, it is inessential.

Proof. of Tutte-Berge Formula.
<u>Goal:</u>Show a matching M and $A \subseteq V$ with exactly $oc(G \setminus A) - |A|$ vertices (which is saying $oc(G \setminus A) - |A| = |V| - 2|M|$). If such M, A are found, then the Tutte-Berge formula is proved. As we have shown before, $\nu(G) \leqslant \frac{1}{2}(|V| - oc(G \setminus A) + |A|), \forall A \subseteq V$, so $\nu(G) \leqslant \frac{1}{2}\min\{|V| - oc(G \setminus A) + |A|\colon A \subseteq V\}$. When $oc(G \setminus A) - |A| = |V| - 2|M|$ many M-exposed vertices, we know

$$\nu(G) \geqslant |M| = \frac{1}{2}(|V| - oc(G \setminus A) + |A|) \leqslant \frac{1}{2}\min\{|V| - oc(G \setminus A) + |A|\colon A \subseteq V\}$$

so the Tutte-Berge Formula holds.
Now we do induction on $m = |E|$
<u>Base:</u> $m = 0$, let $A = \emptyset$, we are done. Now assume $m \geqslant 1$ and pick $uv \in E$:

Case 1: v is essential. Let $G' = G \setminus v$, then $\nu(G') < \nu(G)$. By induction, there exists matching M' in G' and $A' \subseteq V \setminus \{v\}$ with

$$|M'| = \frac{1}{2}(n - 1 - oc(G' \setminus A') + |A'|)$$

Let M be a matching of G with $|M| = \nu(G)$. Pick $e \in \delta(v) \cap M$ (it exists by v being essential). Then $\overline{M} = M \setminus e$ is a matching in G' which implies $|\overline{M}| = |M| - 1 \leqslant |M'|$. Now, suppose $|M| - 1 < |M'|$, then $|M| \leqslant |M'|$, then since M' is also a matching in G, we have $|M| \geqslant |M'|$, so $|M| = |M'|$, then M' is a maximum cardinality matching in G without v, so v is not essential, contradiction. Hence, $|M| - 1 = |M'|$. Then let $A = A' \cup \{v\}$, then $|A| = |A'| + 1$ and $G \setminus A = G' \setminus A'$, so

$$|M| = |M'| + 1 = \frac{1}{2}(n + 1 - oc(G' \setminus A') + |A'|) = \frac{1}{2}(n - oc(G \setminus A) + |A|)$$

we are done.

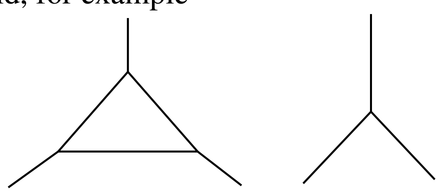
Case 2: u, v both are inessential. Later.

Let C be an odd cycle, let $G' = G/C$ (contracting C). That is $V'(G) = V(G) \setminus V(C) \cup \{\mathcal{C}\}$; $E(G') = \{e \in E(G) : e \cap C = \emptyset\} \cup \{v\mathcal{C} : \exists uv \in E(G), u \in V(C), v \notin V(C)\}$. Note from this point, we allow parallel edges. The idea is that a matching in G' can be extend to a matching in G with the same number of exposed vertices. The process is, let all edges in the matching of G' be in the matching of G, then let one vertex in C to represent the $\mathcal{C}$ in G', and C has even number of vertices left, then choose edges so they are all M-covered.

Proposition 38

Let $G = (V, E)$, C an odd cycle, $G' = G/C$. Let M' a matching in G'. Then there exists a matching M of G such that the number of M-exposed vertices in G equals the number of M'-exposed vertices in G'.

Note we add $\frac{|C|-1}{2}$ new edges to M' to get M. Therefore, $\nu(G) \geqslant \nu(G') + \frac{c-1}{2}$, but the equality does not necessarily hold, for example



where the left graph G has $\nu(G) = 3$, the right one has $\nu(G') = 1$ and $\frac{|C|-1}{2} = 1$. An odd cycle is tight if $\nu(G) = \nu(G') + \frac{|C|-1}{2}$.

Now back to the proof, we pick a tight cycle C containing uv and where $\mathcal{C}$ is inessential in $G' = G/C$. Then there exist M' matching of G', $A' \subseteq V(G')$:

$$|M'| = \frac{1}{2}(|V(G') - oc(G' \setminus A') + |A'|)$$

If $\mathcal{C} \notin A'$, then any component of $G' \setminus A'$ containing $\mathcal{C}$ will be a component of $G \setminus A$ of same pairing after extending back (that is, if the component in G' is odd, then the component in G will also be odd because there are even number of vertices if deleting $\mathcal{C}$, and C has odd number of vertices, same if the component in G' is even). Hence, there are

$$oc(G' \setminus A') - |A'| = oc(G \setminus A) - |A| = |V| - |C| + 1 - 2|M'| = |V| - |C| + 1 - 2\left(|M| - \frac{|C|-1}{2}\right) = |V| - 2|M|$$

many M-exposed vertices.

Q: But why does such C exist? What if $\mathcal{C} \in A'$? $\square$

3.4 Matching 4

Lemma 39

Let $uv \in E$. If u, v are inessential, then there is a tight odd cycle C containing the edge uv, such that $\mathcal{C}$ is inessential in $G' = G/C$.

Proof. Let M_u, M_v be maximum cardinality matchings exposing u, v respectively. (Note1: $uv \notin M_u \cup M_v$; Note 2: Mu, Mv covers v, u respectively by the maximality). Then

- Degree of u, v is 1 in $M_u \triangle M_v := F$ ((V, F) is a vertex disjoint union of M_u, M_v alternating paths/cycles).
- There exists an alternating path P starting at u and the other end z is M_v-exposed. Suppose the other end is M_u-exposed, then the path P is an M-augmenting path, contradicts to the maximality of M in G. If $z \neq v$, then $vu + P$ is an M_v augmenting path in G, contradiction. Hence, P is an alternating path from u to v, let $C = uv + P$, note C is an odd cycle because P has even length (by $v = z$ is M_v exposed).
  - $\delta(C) \cap M_u = \emptyset$. Since the path is alternating, the only vertex in C not incident to a M_u edge in C is u, but since u is M_u-exposed, $\delta(u) \cap M_u) = \emptyset$.
  - $M_u \setminus C$ is a maximum cardinality matching in $G \setminus C$. Suppose not, then there is a larger matching M' in $G \setminus C$. And consider $M' \cup \{M_u \cap E(C)\}$, it is a matching in G because $\delta(C) \cap M_u = \emptyset$. And it is larger matching in G than M_u because

    $$|M_u| = |M_u \cap E(G \setminus C)| + |M_u \cap E(C)| < |M'| + |M_u \cap E(C)| = |M' \cup \{M_u \cap E(C)\}|$$

    contradiction. Hence, $\mathcal{C}$ is inessential in G'.
  - Hence, $M_u \setminus C$ is a maximum cardinality matching in G/C without including $\mathcal{C}$, so $\mathcal{C}$ is inessential in G/C. Since there are $\frac{|C|-1}{2}$ many M_u vertices in C, we know

    $$\nu(G) = |M_u| = |M_u \setminus C| + |M_u \cap E(C)| = \nu(G/C) + \frac{|C|-1}{2}$$

    so C is a tight odd cycle containing uv, as required.

Lemma 40

Let M be a matching, $A \subseteq V$ such that $|M| = \frac{1}{2}(|V| - oc(G \setminus A) + |A|)$. Then all vertices in A are essential.

Proof. Let $v \in A$. Let $A' = A \setminus \{v\}, V' = V \setminus \{v\}, G' = G \setminus \{v\}$. Since the components of $G \setminus A$ are the same as the components of $G' \setminus A'$, we know

$$\begin{aligned} oc(G \setminus A) &= oc(G' \setminus A') \\ \nu(G') &\leqslant \frac{1}{2}(|V'| - oc(G' \setminus A') + |A'|) \\ &= \frac{1}{2}(|V| - 1 - oc(G \setminus A) + |A| - 1) \\ &= |M| - 1 \end{aligned}$$

so v is essential. □

Then answer our question, $\mathcal{C} \in A'$ reaches a contradiction because $\mathcal{C}$ is inessential. Hence, such C exists, and $\mathcal{C} \notin A'$. □

3.5 Matching 5

We say an M-alternating tree T is <u>frustrated</u> if $\forall uv \in E$, $u \in B(T)$, we have $v \in A(T)$.

Proposition 41

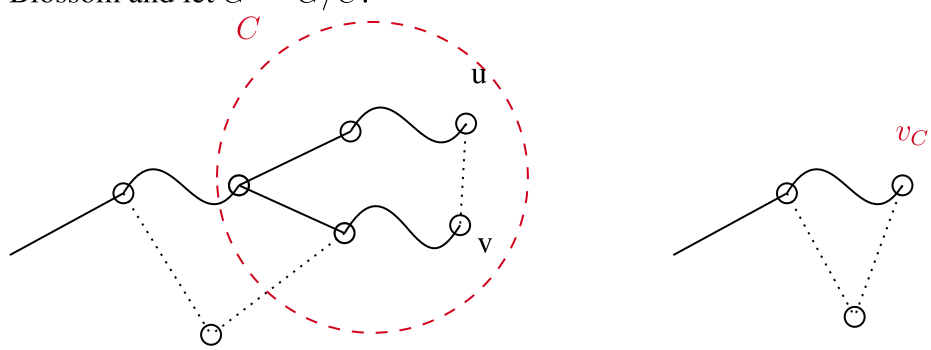
If T is frustrated, then G has no perfect matching.

Proof. Since all neighbors of vertices in $B(T)$ are in $A(T)$, we know $G \setminus A(T)$ has at least $|B(T)|$ many odd components, because each vertex in $B(T)$ in $G \setminus A(T)$ is an odd component. Hence,

$$|oc(G \setminus A(T))| \geqslant |B(T)| > |A(T)|$$

then by Tutte's Matching Theorem, we know G has no perfect matching. □

Let $u, v \in B(T)$ such that $uv \in E$, then $T + uv$ has a unique odd cycle C (called Blossom). Srhink the Blossom and let $G' = G/C$.



Note:

- Edges in $M \setminus E(C)$ form a matching M' in G'.
- Shrunken Tree T' is M'-alternating in G'.
- Psuedonode v_C is in the set $B(T')$ for the tree T.

Note: One may need to shrink multiple times.
We say the graph obtained after shrinking (sequentially) Blossoms is a <u>derived graph</u>.
$S(v)$ will represent the set of vertices that have been shrunk into $v \in V(G')$, then

$$\forall v \in V(G'), S(v) = \begin{cases} v, & \text{if } v \in V(G) \\ \cup_{w \in C} S(w), & \text{if } v = v_C, \text{ for some Blossom } C \end{cases}$$

Note: $|S(v)|$ is odd, $\forall v \in V(G')$ by definition $|S(v)| = 1$ or it's a sum of odd many odd numbers.

Proposition 42

Let G' be a derived graph from G, M' a matching of G', T' an M'-alternating frustrated tree of G' with all pseudonode in $B(T')$, then G has no perfect matching.

Proof. If G has a perfect matching M, then for any Blossom C, G/C also has a perfect matching $M \setminus C$, hence, G' will have a perfect matching, but G' has an M'-alternating frustrated tree,contradiction. $\square$

Proposition 43

Let G' be derived graph from G, M' an matching of G', T' an M'-alternating tree, $uv \in E(G')$ with $u, v \in B(T')$, C' unique cycle (Blossom) in $T' + uv$.
Then $M'' = M' \setminus E(C')$ is a matching for $G'' = G'/C'$ and $T'' = (V(T') \setminus V(C') \cup \{v_{C'}\}, E(T') \setminus E(C'))$ is an M''-alternating tree in G'' with $v_{C'} \in B(T'')$.

Algorithm 7 Blossom Algorithm for Perfect Matching

```
Input graph G and matching M of G
Set M' = M, G' = G
Choose an M'-exposed node r of G' and put T = ({r}, ∅)
while there exists vw ∈ E' with v ∈ B(T), w ∉ A(T) do
    if w ∉ V(T), w is M'-exposed then
        Use vw to augment M'
        Extend M' to a matching M of G
        Replace M' by M and G' by G
        if there is no M'-exposed node in G' then
            Return the perfect matching M' and stop
        else
            Replace T by ({r}, ∅) where r is M'-exposed.
        end if
    else if w ∉ V(T), w is M'-covered then
        Use vw to extend T
    else if w ∈ B(T) then
        Use vw to shrink and update M' and T
    end if
end while
return G', M', T and stop; G has no perfect matching.
```

Theorem 44

Blossom algorithm does $O(n)$ augmentation, $O(n^2)$ shrinks, $O(n^2)$ tree extensions and correctly determines if G has perfect matchings.

Proof. Each augmentation increase $|M'|$ by 1, implies $O(n)$ augmentation. Between two augmentation steps, shrink reduces size of G' by at least 2 vertices implies $O(n)$ shrinks, so total $O(n^2)$ shirnks. Similar for tree extensions. $\square$

Algorithm 8 Blossom Algorithm for Maximum Cardinality Matching

Input graph G and matching M of G
Set $M' = M$, $G' = G$, $\mathcal{T} = \emptyset$
($\star$) Choose an M'-exposed node r of G' and put $T = (\{r\}, \emptyset)$
while there exists $vw \in E'$ with $v \in B(T), w \notin A(T)$ **do**
  if $w \notin V(T)$, w is M'-exposed **then**
    Use vw to augment M'
    Extend M' to a matching M of G
    Replace M' by M and G' by G
    if there is no M'-exposed node in G' **then**
      Return the perfect matching M' and stop
    else
      Replace T by $(\{r\}, \emptyset)$ where r is M'-exposed.
    end if
  else if $w \notin V(T)$, w is M'-covered **then**
    Use vw to extend T
  else if $w \in B(T)$ **then**
    Use vw to shrink and update M' and T
  end if
end while
$\mathcal{T} \leftarrow \mathcal{T} \cup \{T\}$; $G' \leftarrow G' \setminus V(T)$; $M' \leftarrow M' \setminus E(T)$
if There exists an M'-exposeed node in G' **then**
  go back to ($\star$)
else
return $M = \cup_{T\in\mathcal{T}} M_T$
end if

Proof. Let $T_1, \ldots, T_k$ be the trees in T; M be the final matching. For each T_i, there exists only one M-exposed vertex in T_i because each T_i is an M_{T_i}-alternating tree, so the only M-exposed vertex in T_i is its root, so there are k M-exposed vertices in total. Let $A = \cup_{i=1}^{k} A(T_i)$. Each vertex in $B(T_i)$ is an odd component of $G \setminus A$ because each T_i is frustrated, all neighbors of vertices in $B(T_i)$ are in A. Hence,

$$oc(G \setminus A) \geqslant \sum_{i=1}^{k} |B(T_i)| \geqslant \sum_{i=1}^{k} (|A(T_i)|+1) = |A|+k$$

which implies

$$|M| = \frac{|V|-k}{2} \geqslant \frac{1}{2}(|V| - oc(G \setminus A) + |A|)$$

so M is a maximum cardinality matching. $\square$

3.6 Matching 6

Definition 45: Gallai-Edmonds Decomposition

Let $G = (V, E)$, B be the set of inessential vertices, $C := \{v \in V \setminus B : v \in N_G(B)\}$, $D := V \setminus (C \cup B)$. (B, C, D) is called the Gallai-Edmonds partition/decomposition of G.

Proposition 46

Let $T_i, i = 1, \ldots, k$ be the frustrated trees found in Blossom algorithm. Then

$$C = \cup_{i=1}^k A(T_i),\ B = \cup_{i=1}^k (\cup_{v \in B(T_i)} S(v)),\ D = V \setminus (B \cup C)$$

Note.

- This implies all components of $G[B]$ are odd and C is a minimizer of Tutte-Berge Formula.
- This also implies that Gallai-Edmonds decomposition can be computed in polytime.
- Implies $G[D]$ only has even components. (every vertex in D is M-covered, and it's not matched to A nor B).

Proof. We saw all vertices in $\cup_{i=1}^k A(T_i)$ are essential (by the proof of correctness of the Blossom Algorithm, we know A is the minimizer of Tutte-Berge Formula hence all vertices in it is essential). For all $v \in \cup_{i=1}^k \left(\cup_{v \in B(T_i)} S(v)\right)$, there exists an even M-alternating path from an M-exposed vertex u to it. Pick such path P, and then $M' = M \triangle E(P)$ is a matching with $|M'| = |M|$, and v is M'-exposed which implies that v is inessential.

- Consider $v \in V \setminus \underbrace{\left(\cup_{i=1}^k A(T_i) \cup \left(\cup_{i=1}^k \left(\cup_{v \in B(T_i)} S(v)\right)\right)\right)}_{D'}$, and consider $G' = G \setminus v$. Since D' only has even components, we know $oc(G' \setminus C) = oc(G \setminus C \setminus v) > oc(G \setminus C)$, not D is not connected to B, so we removing v will not increas the number of components in B, but only D. Hence

$$\nu(G') \leqslant \frac{1}{2}(|V| - 1 - oc(G' \setminus C) + |C|) < \frac{1}{2}(|V| - oc(G \setminus C) + |C|) = \nu(G)$$

- Hence, v is essential.

Note.

- $v \in D'$ is not adjacent to a vertex in B, otherwise, if v is M-covered, we can extend M, if it's M-exposed, we can augment M.
- $v \in C$ is adjacent to a vertex in B by the definition of the alternating trees.

$\square$

4 Weighted Matching

4.1 Weighted Matching 1

Minimum Weight Perfect Matching
Given $G = (V, E)$, $c_e \in \mathbb{R}, \forall e \in E$, find a perfect matching M of G minimizing $c(M) = \sum_{e \in M} c_e$.
Idea:

$$\begin{aligned} \min \sum_{e \in E} c_e x_e \\ s.t. \quad x(\delta(v)) = 1, \forall v \in V \\ x \geqslant 0, x \in \mathbb{Z}^E \end{aligned}$$

and we can have the relaxation as

$$\begin{aligned} (P_M):\ min \quad & \sum_{e \in E} c_e x_e \\ \text{s.t.} \quad & x(\delta(v)) = 1, \forall v \in V, \\ & x \geqslant 0 \end{aligned}$$

$$\begin{aligned} (D_M):\ max \quad & \sum_{v \in V} y_v \\ \text{s.t.} \quad & y_u + y_v \leqslant c_{uv}, \forall uv \in E \end{aligned}$$

Note: Z_{P_M} := optimal value of (P_M), so $Z_{P_M} \leqslant c(M), \forall$ perfect matching M. (Notice that every perfect matching's indicator vector is a feasible solution for P_M).
Q: Can we solve our problem by solving (P_M)?

Every perfect matching has at least one edge with cost 1. However, the optimal value of P_M is 0 because we can give 0.5 to those edges of the triangles.

Theorem 47: Birkhoff

Let $G = (V, E)$ be bipartite, $c \in \mathbb{R}^E$, then G has a perfect matching if and only if (P_M) is feasible. Moreover, if (P_M) is feasible, then let M^* be a minimum cost perfect matching, then we have $Z_{P_M} = c(M^*)$.

Proof.

$$G \text{ has a perfect matching} \iff P_M \text{ is feasible (SKIPPED)}$$

Remaining statement: Algorithmic Proof.
Construct a matching H that corresponds to a optimal solution to (P_M) using Complementary Slackness:

- Let $\overline{y}$ be feasible for (D_M).
- Let $E^= := \{uv \in E : \overline{y}_u + \overline{y}_v = c_{uv}\}$
- If $G^= := (V, E^=)$ has a perfect matching M, then $x^M, \overline{y}$ satisfy Complementary Slackness conditions, so we are done, we know M is a minimum weighted perfect matching.
- Else, update $\overline{y}$.

But how should we update $\overline{y}$?
Recall at the end of the algorithm for perfect matching on $G^=$, we will be in one of the two situations

a) Found a perfect matching M, and it's the min weighted perfect matching in G.

b) It finds a frustrated tree in $G^=$.

Idea: Update $\overline{y}_v' s$ to get $E^=_{new}$ such that

- $\overline{y}$ is still feasible for (D_M).
- Current $M \subseteq E^=_{new}$.
- Current $E(T) \subseteq E^=_{new}$.
- At least one edge $uv \in E \setminus E^=_{old} : u \in B(T), v \in V(T)$ is in $E^=_{new}$.

Let $\epsilon = \min\{c_{uv} - \overline{y}_u - \overline{y}_v : u \in B(T), v \notin V(T)\}$, and let $\overline{y}^*_u = \begin{cases} \overline{y}_u + \epsilon, & \forall u \in B(T) \\ \overline{y}_u - \epsilon, & \forall u \in A(T) \\ \overline{y}_i, & \forall u \notin V(T) \end{cases}$.

- $\overline{y}^*$ is still feasible for (P_M). Since the graph is bipartite, no $uv \in E$ such that $u, v \in B(T)$. If $u \in B(T), v \in A(T)$, then $\overline{y}^*_u + \overline{y}^*_v = \overline{y}_u + \overline{y}_v$. If $u \in A(T)$, $v \notin V(T)$, then $\overline{y}^*_u + \overline{y}^*_v = \overline{y}_u + \overline{y}_v - \epsilon \leqslant \overline{y}_u + \overline{y}_v \leqslant c_{uv}$. If $u \in B(T), v \notin V(T)$, then

$$\overline{y}^*_u + \overline{y}^*_v = \overline{y}_u + \overline{y}_v + \epsilon \leqslant \overline{y}_u + \overline{y}_v + c_{uv} - (\overline{y}_u + \overline{y}_v) = c_{uv}$$

If $u, v \notin V(T)$, then $\overline{y}^*_u + \overline{y}^*_v = \overline{y}_u + \overline{y}_v$.

- $M \subseteq E^{=}_{new}$. Consider any edge uv in $E(T)$, then it has one end in $A(T)$ and the other end in $B(T)$, so we know $\overline{y}^*_u + \overline{y}^*_v = \overline{y}_u + \overline{y}_v = c_{uv}$, so $uv \in E^{=}_{new}$. That is, $M \subseteq E(T) \subseteq E^{=}_{new}$.

$\square$

Algorithm 9 Min Weight Perfect Matching Algorithm for Bipartite Graphs

Let y be a feasible solution to (P_M), M a matching of $G^{=}$
If M is a perfect matching of G, return M and stop
Set $T \leftarrow (\{r\}, \emptyset)$ where r is an M-exposed node of G
while not stopped **do**
    while there exists $vw \in E^{=}$ with $v \in B(T)$, $w \notin V(T)$ **do**
        if w is M-exposed **then**
            Use vw to augment M
            if there is no M-exposed node in G **then**
                Return the perfect matching and stop
            else
                Replace T by $(\{r\}, \emptyset)$ where r is M-exposed
            end if
        else
            Use vw to extend T
        end if
    end while
    if every $vw \in E$ with $v \in B(T)$ has $w \in A(T)$ **then**
        Stop, G has no perfect matching
    else
        Let $\epsilon = \min\{c_{vw} - y_v - y_w : v \in B(T), w \notin V(T)\}$
        Replace y_v by $y_v + \epsilon$ for $v \in B(T)$, $y_v - \epsilon$ for $v \in A(T)$.
    end if
end while

Note.

- $M \subseteq E^{=}$ all the time, so if we find a perfect matching, it will be a min weight one.
- Stopping points: either we find a perfect matching in G or we find a frustrated tree in G (not $G^{=}$), so there is no perfect matching.
- The loop can only run polynomially many times because for every iteration, one more egde will be added to T, so the algorithm will terminate in polynomial time.

4.2 Weighted Matching Two

Rather than the P_M we used in the previous subsection, we now consider the fact that if $S \subseteq V$ and $|S|$ is odd and $|S| \geqslant 3$, any perfect matching must use at least one edge in $\delta(S)$ because if we pick the pairs of two from S, there is always one left, and we let $|S| \geqslant 3$ because if $|S| = 1$, then the condition is necessary for having a perfect matching. Let $\vartheta = \{S : S \subseteq V, |S| \text{ is odd and } |S| \geqslant 3\}$. We introduce the new problem

$$
\begin{aligned}
(P'_M)\ min \quad & \sum_{e\in E} c_e x_e \\
\text{s.t.} \quad & x(\delta(v)) = 1, \forall v \in V, \\
& x(\delta(S)) \geqslant 1, \forall S \in \vartheta, \\
& x \geqslant 0
\end{aligned}
$$

$$
\begin{aligned}
(D'_M)\ max \quad & \sum_{v\in V} y_v + \sum_{S\in\vartheta} y_S \\
\text{s.t.} \quad & y_u + y_v + \sum_{S\in\vartheta : uv\in\delta(S)} y_S \leqslant c_{uv}, \forall uv \in E, \\
& y_S \geqslant 0 \quad , \forall S \in \vartheta
\end{aligned}
$$

Note. When needed, will use $P'_M(G), D'_M(G)$ to explicitly refer to which graph is being used.

Define $\overline{c}_{uv} = c_{uv} - \sum_{S\in\vartheta:uv\in\delta(S)} y_S - y_u - y_v$.

Complementary Slackness Conditions:

- $x(\delta(v)) = 1$ which is satisfied by any perfect matching
- $x(\delta(S)) = 1$ or $y_S = 0$
- $x_{uv} = 0$ or $\overline{c}_{uv} = 0$.

We want to construct perfect matching M with

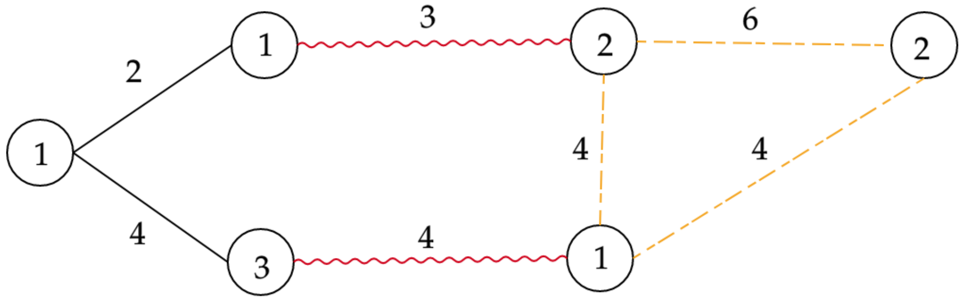
$$\overbrace{M \subseteq E^{=} = \{e \in E : \overline{c}_e = 0\}}^{(*)} \text{ AND } \underbrace{|M \cap \delta(S)| = 1, \forall S \in \vartheta : y_S > 0}_{(**)}.$$

when needed, we use $E^{=}(G, \overline{y})$.

Basic Algorithm Sketch

- Start with $\overline{y} : \overline{y}_S = 0, \forall S \in \vartheta$, feasible for (D'_M).
- If found a perfect matching M in $G^{=}$, we are done.
- Else, look at the frustrated tree T in $G^{=}$ and update $\overline{y}$ so that more vertices can be added to T.

Let's look at the example below. Assume $y_S = 0, \forall S \in \vartheta$ and the value of y_v is given for every $v \in V$.



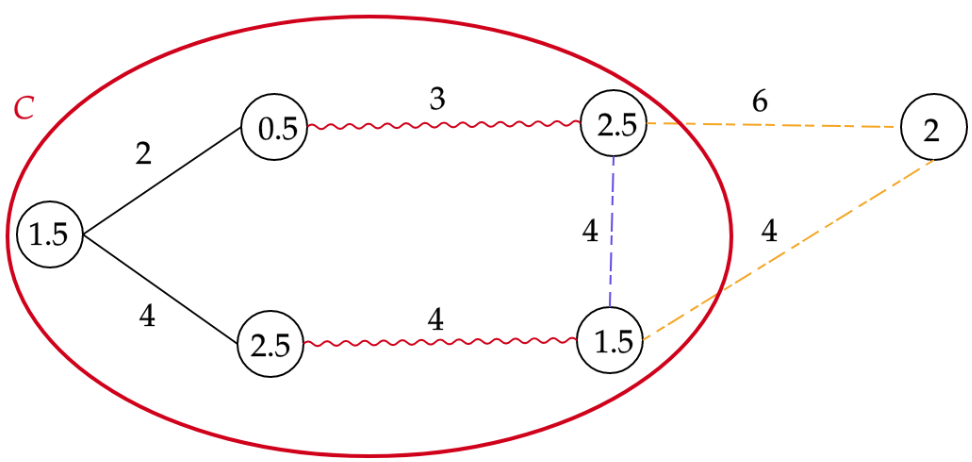
where red curved lines means the edges in M and the yellow dashed ones means the edges in E but not in $E^{=}$.
If we let $\epsilon_1 = \min\{\bar{c}_{uv} : u \in B(T), v \notin V(T)\}$, then we have $\epsilon_1 = \min\{6-2-2, 4-2-1\} = 1$. However, if we update $\overline{y}_u$ by

$$\overline{y}_u = \begin{cases} \overline{y}_u + \epsilon_1, & \forall u \in B(T) \\ \overline{y}_u - \epsilon_1, & \forall u \in A(T) \\ \overline{y}_i, & \forall u \notin V(T) \end{cases}$$

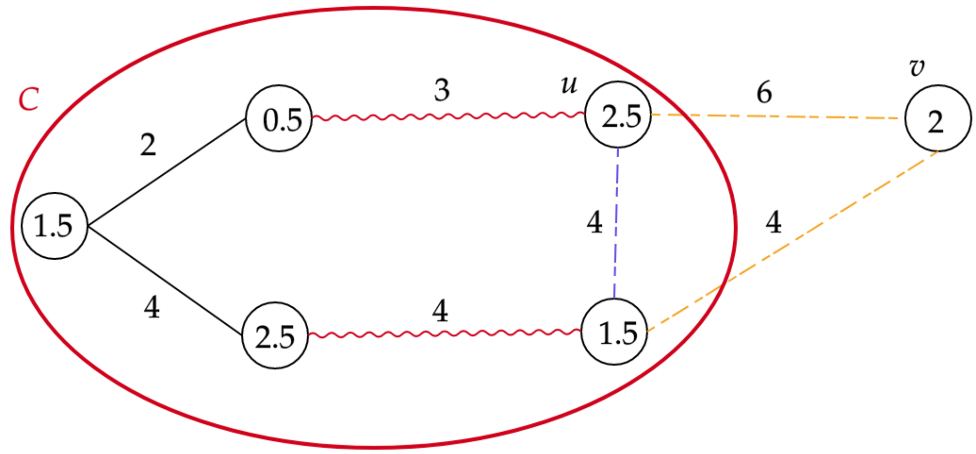
, we realized we have $\bar{c}_{uv} = 4 - 3 - 2 = -1$ for the $u, v \in B(T) \setminus r$. So we also introduce $\epsilon_2 = \min\{\bar{c}_{uv}/2 : u \in B(T), v \in B(T)\}$ and $\epsilon = \min\{\epsilon_1, \epsilon_2\}$. Then update

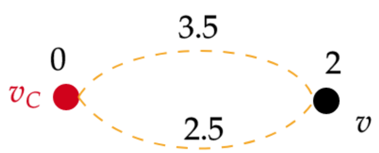
$$\overline{y}_u = \begin{cases} \overline{y}_u + \epsilon, & \forall u \in B(T) \\ \overline{y}_u - \epsilon, & \forall u \in A(T) \\ \overline{y}_i, & \forall u \notin V(T) \end{cases}$$

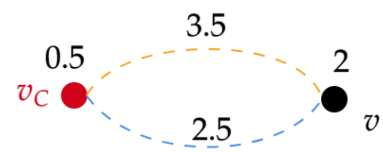
In the above example, $\epsilon = \epsilon_2 = 1/2$. Now we have



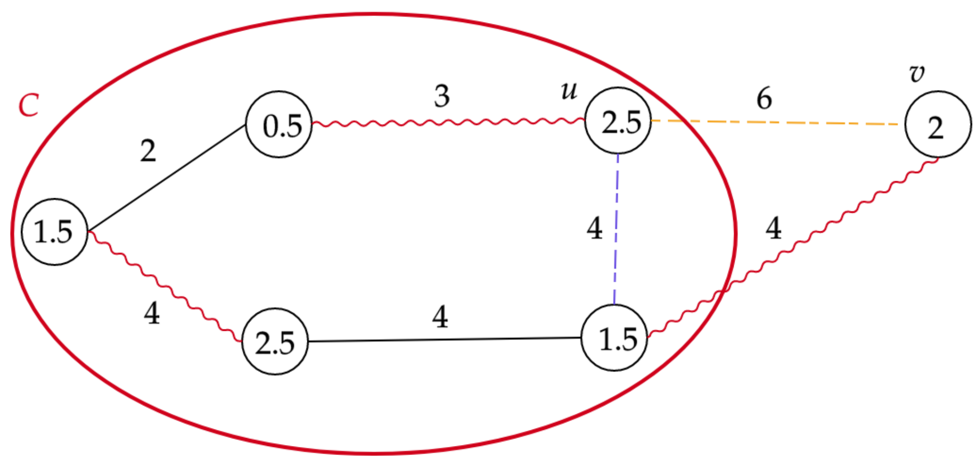
Notice the blue dashed edge means the edge is in $E^=$. Then, we realize that we find a blossom in $G^=$. If we find a perfect matching in $G^=/C$, then it can be extended to a perfect matching of $G^=$ satisfying $(**)$. But how to shrink? Notice that $\bar{c}_{uv} = c_{uv} - y_u - y_v - \sum_{S\in\vartheta:uv\in\delta(S)} y_S$, so we can change y_S (here, y_C). And define the parallel edges as $c_{v_c v} = c_{uv} - y_u$. Then adjust y_{v_C} accordingly, get

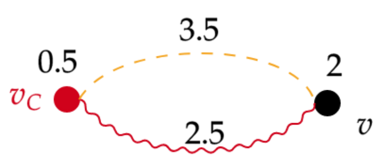






Then we find a matching M satisfying both $(*)$ and $(**)$.





Shrinking a blossom C:
We say G', c' is **derived** from G, c by shrinking blossom C if G', c' are defined as:

- $V(G') = (V(G) \setminus C) \cup \{v_c\}$
- For every $uv \in E$,
  - If $u, v \notin V(C)$, add uv to $E(G')$, let $c'_{uv} = c_{uv}$.
  - If $u \in V(C), v \notin V(C)$, add $v \notin V(C)$, add $v_c v$ to $E(G')$ with cost $c'_{v_c v} = c_{uv} - y_u$.

Now we have Basic Algorithm Sketch:

- Start with $\overline{y} : \overline{y}_S = 0, \forall S \in \vartheta$, feasible for (D'_M).
- If found a perfect matching M in $G^=$, we are done.
- Else, look at the frustrated tree T in $G^=$ and update $\overline{y}$ so that more vertices can be added to T. If update allows to find blossom C, with $E(C) \subseteq E^=$, shrink it, set $y_{v_c} = 0$.

Proposition 48

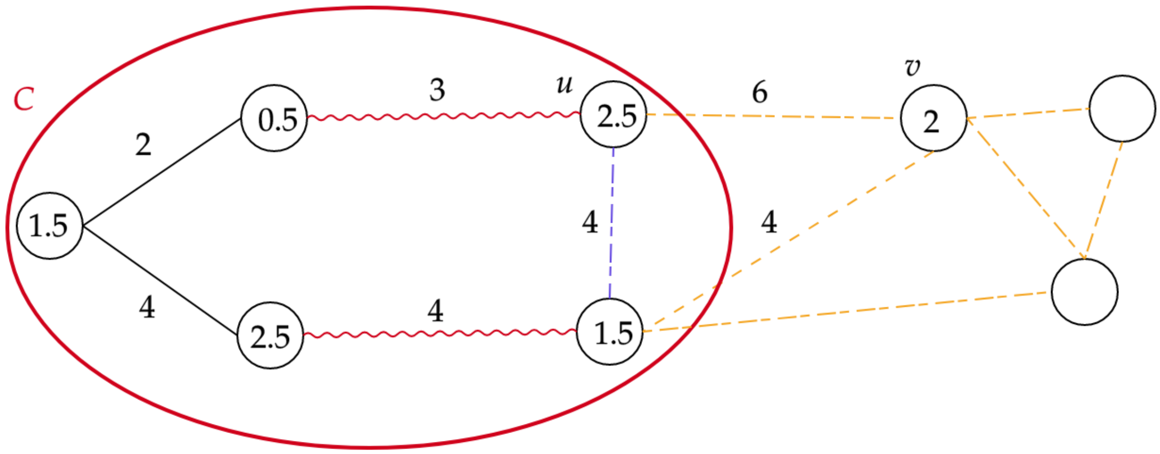
Let $\overline{y}$ be feasible for $D'_M(G)$, with $\overline{y}_S = 0, \forall S \in \vartheta(G)$. Let G', c' be derived from G, c by shrinking blossom C with $E(C) \subseteq E^=(G, \overline{y})$. Let M' be a perfect matching of G' and y' feasible for $D'_M(G')$, where M', y' satisfy $(*), (**)$ and $y'_{v_c} \geqslant 0$. Then extend M' tp a perfect matching $\hat{M}$ of G and define $\hat{y}$ as

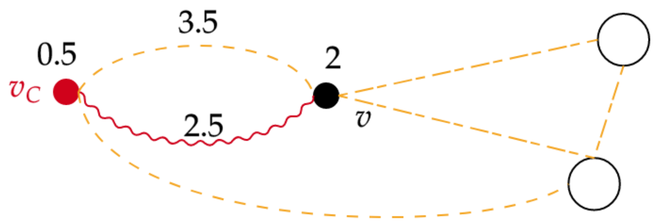
- $\hat{y}_v = \overline{y}_v, \forall v \in V(C)$.
- $\hat{y}_v = y'_v, \forall v \in V(G') \setminus v_c$.
- $\hat{y}_{S(v_c)} = y'_{v_c}$
- $\hat{y}_{S(D)} = y'_D, \forall D \in \vartheta(G')$
- $\hat{y}_S = 0$, for every other $S \in \vartheta$.

Then $\hat{y}$ is feasible for $D'_M(G)$ and $\hat{M}, \hat{y}$ satisfy $(*), (**)$.

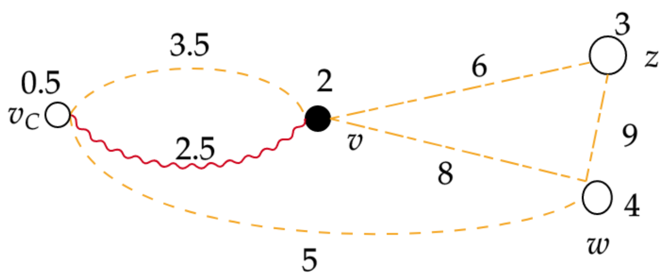
4.3 Weighted Matching Three

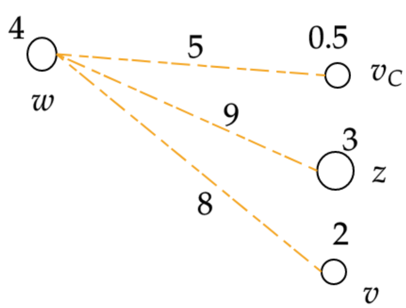
Now, what if we don't find a perfect matching in G'?

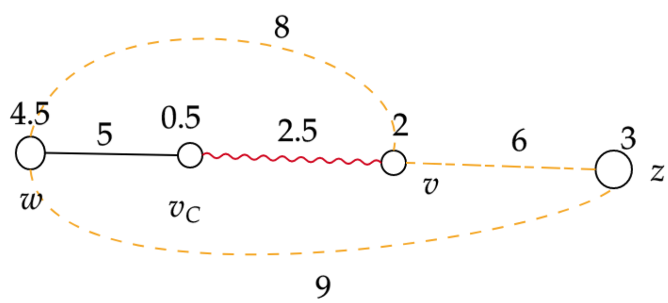




If we start working back in G, we need to keep track of all $y_S > 0$, that's not good. Instead, we keep working on G'!



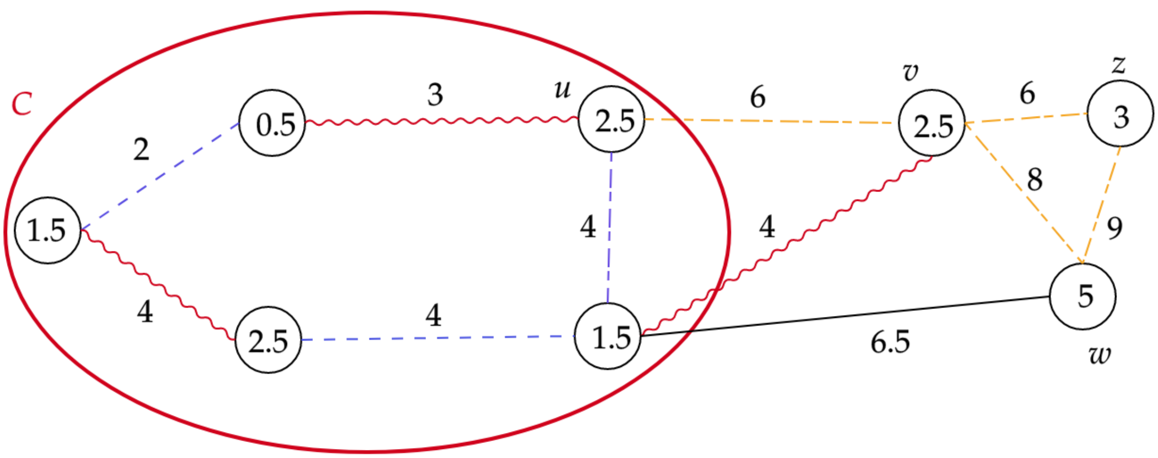


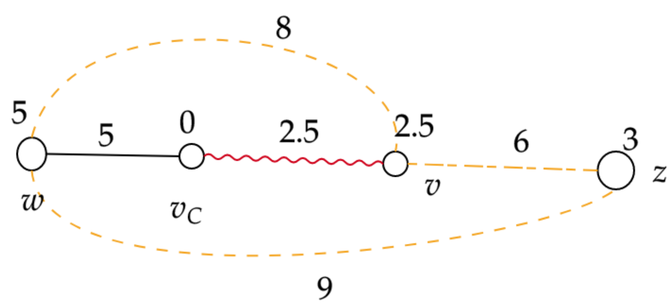


Note we pick w as r and change y_w. Now we have

$$\begin{aligned}
\epsilon_1 &= \min\{\overline{c}_{uv} : u \in B(T), v \notin V(T)\} \\
\epsilon_2 &= \min\{\overline{c}_{uv}/2 : u \in B(T), v \notin V(T)\} \\
\epsilon &= \min\{\epsilon_1, \epsilon_2\} \\
\overline{y}_u &= \begin{cases} \overline{y}_u + \epsilon, & \forall u \in B(T) \\ \overline{y}_u - \epsilon, & \forall u \in A(T) \\ \overline{y}_i, & \forall u \notin V(T) \end{cases}
\end{aligned}$$

In the example, we see that $\epsilon_1 = 1$ and $\epsilon_2 = 0.75$, so $\epsilon = 0.75$. But then if we update with ϵ, we have $\overline{y}_{v_c} = -0.25$, so it's infeasible for D'_M. So we also define $\epsilon_3 = \min\{\overline{y}_u : u \in A(T)$ and u is a pseudonode$\}$. And $\epsilon = \min\{\epsilon_1, \epsilon_2, \epsilon_3\}$. So the new $\epsilon = 0.5$, update $\overline{y}$, we are stuck, so expand the v_C back and for each edge uv with $u \in V(C)$, $v \notin V(C)$ replace c'_{uv} by $c'_{uv} + \overline{y}_u$.





Now let's consider the algorithm

Algorithm 10 Blossom Algorithm for Minimum-Weight Perfect Matching

Let y be a feasible solution to (D'_M), M' a matching of $G^=$, $G' = G$.
If M' is a perfect matching of G, return M' and stop
Set $T \leftarrow (\{r\}, \emptyset)$ where r is an M'-exposed node of G
while not stopped **do**
  while there exists $vw \in E^=$ with $v \in B(T)$, $w \notin V(T)$ **do**
    if w is M'-exposed **then**
      Use vw to augment M
      if there is no M'-exposed node in G **then**
        Return the perfect matching and stop
      else
        Replace T by $(\{r\}, \emptyset)$ where r is M'-exposed
      end if
    else
      Use vw to extend T
    end if
  end while
  if There exists $uv \in E^=$ **then**
    Use uv to shrink and update M', T and c'
  else if every $vw \in E$ with $v \in B(T)$ has $w \in A(T)$ and $A(T)$ contains no pseudonode **then**
    Stop, G has no perfect matching
  else if There is a pseudonode $v_c \in A(T)$ with $y_{v_c} = 0$ **then**
    Expand v and update M', T, and c'
  else

$$\epsilon_1 = \min\{\bar{c}_{uv} : u \in B(T), v \notin V(T)\}$$
$$\epsilon_2 = \min\{\bar{c}_{uv}/2 : u \in B(T), v \notin V(T)\}$$
$$\epsilon_3 = \min\{\bar{y}_u : u \in A(T) \text{ and } u \text{ is a pseudonode}\}\epsilon \quad = \min\{\epsilon_1, \epsilon_2, \epsilon_3\}$$
$$\bar{y}_u = \begin{cases} \bar{y}_u + \epsilon, & \forall u \in B(T) \\ \bar{y}_u - \epsilon, & \forall u \in A(T) \\ \bar{y}_i, & \forall u \notin V(T) \end{cases}$$

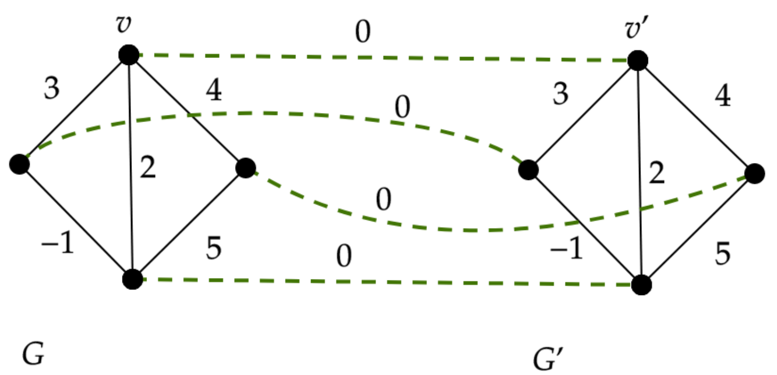
  end if
end while

Correctness can be shown by argueing each step preserves matching $\subseteq E^=$. Polytime: Bound number of steps while keeping same matching. Note: Shrink and unshrink may lead to infinite loop.

4.4 Maximum Weight Matching

Find matching M maximizing $c(M)$. We can use max. weighted perfect matching problem.
Let $G = (V, E)$, $c : E \mapsto \mathbb{R}$. Let G' be a copy of G with exact same edge costs (if $v \in V(G)$, v' is corresponding copy). Let $\overline{G}$ be graph with vertices $V(G) \cup V(G')$ edges $E(G) \cup E(G') \cup$

$\underbrace{\{vv' : v \in V(G)\}}_{c(vv')=0, \forall v \in V(G)}$.



Clearly, $\overline{G}$ has a perfect matching.

Proposition 49

Let $\overline{M}$ be a max. weighted perfect matching in $\overline{G}$. Then $M = \overline{M} \cap E(G)$ is a max. weighted matching in G.

Proof. It is clear that M is a matching of G. Let M^* be a max weighted matching of G. Let U =set of M^*-exposed vertices in G.

- First, we can construct a perfect matching M'' in $\overline{G}$ by copying M^* in G', and for $v \in U$, we include $vv' \in M''$, so $c(M'') = 2c(M^*) + c(E(U, U')) = 2c(M^*) + 0$.
- Let $M' = \overline{M} \cap E(G')$, we know $c(M) = c(M')$ because if say $c(M) > c(M')$, then by previous point, we create a perfect matching with larger weight, contradiction.
- So $2c(M^*) = c(M'') \leqslant c(\overline{M}) = 2c(M)$ implies $c(M^*) \leqslant c(M)$, and by definition, $c(M) \leqslant c(M^*)$, so $c(M) = c(M^*)$, we are done.

□

5 Matroid Intersection

5.1 Matroid Intersection One

Definition 50

Let $M_1 = (S, \mathcal{I}_1)$, $M_2 = (S, \mathcal{I}_2)$ be two matroids over the same ground set S.

- Matroid Intersection Problem (unweighted):
  - Find $A \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $|A|$.
- Matroid Intersection Problem (weighted):
  - Given $c \in \mathbb{R}_+^S$, find $A \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing the cots $c(A)$.

Example 51

Let $G = (V, E)$ be bipartite with bipartition V_1, V_2, let

$$\mathcal{I}_1 = \{A \subseteq E : |A \cap \delta(v)| \leqslant 1, \forall v \in V_1\}$$
$$\mathcal{I}_2 = \{A \subseteq E : |A \cap \delta(v)| \leqslant 1, \forall v \in V_2\}$$

In Example 13, we showed that $(E, \mathcal{I}_1)$ and $(E, \mathcal{I}_2)$ are both matroids. Then the problem of finding $A \in \mathcal{I}_1 \cap \mathcal{I}_2$ maximizing $|A|$ is the same as finding the maximum cardinality matching in G.

We first find the upper bound for the set we can find in the matroid intersection. Let $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and let $A \subseteq S$ be arbitrary.
Goal: Try to account for size of J by splitting it into an M_1 component and an M_2 component.

$$|J| = |\underbrace{J \cap A}_{\in \mathcal{I}_1}| + |\underbrace{J \cap \overline{A}}_{\in \mathcal{I}_2}| \leqslant r_1(A) + r_2(A)$$

where $\overline{A}$ represents the compliment of A in S. Note $J \cap A \in \mathcal{I}_1$ because $J \in \mathcal{I}_1 \cap \mathcal{I}_2$, so its subset is also in $\mathcal{I}_1$, same for the $\mathcal{I}_2$ one.

Theorem 52: Matroid Intersection Theorem (Edmonds 71)

Let $M_i = (S, \mathcal{I}_i), i = 1, 2$ be matroids, then

$$\max\{|J| : J \in \mathcal{I}_1 \cap \mathcal{I}_2\} = \min\{r_1(A) + r_2(\overline{A}) : A \subseteq S\}$$

Now back to Example 51 above. Let M be an maximum cardinality matching of G. By Matroid Intersection Theorem, there exist $A \subseteq E$ such that $|M| = r_1(A) + r_2(\overline{A})$. Let B_1 be an M_1-basis of A, B_2 be an M_2 basis of $E \setminus A = \overline{A}$. Let $U_i = V(B_i) \cap V_i, i = 1, 2$.

- Since $|B_1|= r_1(A)$, and $B_1 \subseteq A$, we know for every $e \in B_1$, it must be incident to one vertex in V_1 and no two edges in B_1 incident to the same vertex in V_1, that is, $r_1(A) = |B_1|= |B_1 \cap V_1|= |U_1|$, similarly, $|U_2|= r_2(E \setminus A)$.

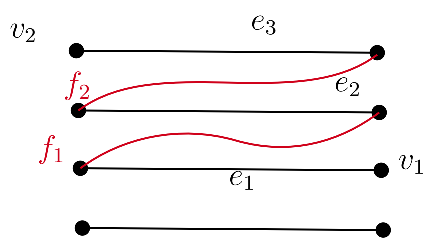
- Consider any edge $uv \in E$, say $u \in V_1, v \in V_2$. Suppose $u \notin U_1, v \notin U_2$, then $uv \notin B_1 \cup B_2$. Notice that if $uv \in A$, then $uv \notin B_1$ implies that there is an edge in B_1 which is incident to u, so $u \in B_1 \cap V_1$, contradiction. That is, $uv \in E \setminus A$, similarly, $uv \notin B_2$ implies that there is an edge in B_2 incident to v, so $v \in U_2$, contradiction. Hence, $U_1 \cup U_2$ is a vertex cover of G.

Theorem 53: König's Theorem

If G is bipartite,

$$\max\{|M|: M \text{ is a matching}\} = \min\{|U|: U \text{ is a vertex cover}\}$$

5.1.1 Matroid Intersection Algorithm:



We see that J (the red curly part) is a matching, and $P = e_1 f_1 e_2 f_2 e_3$ is an augmenting path. Note $e_i \notin J, \forall i = 1, 2, 3$ and $f_i \in J, \forall i = 1, 2$.
How can I write the condition v_1, v_2 are J-exposed, only based on M_1, M_2?

$$J \cup \{e_1\} \in \mathcal{I}_1, J \cup \{e_2\} \in \mathcal{I}_2$$

because e_1 is incident to v_1, $J \cup \{e_1\} \in \mathcal{I}_1$ implies that $J \cup \{e_1\} \cap \delta(v_1) \leqslant 1$, and if $J \cap \delta(v_1) > 0$, then the above inequality does not hold, same for v_2.
How can we make sure it is still a matching after symmetric difference between J and P?

- $J \cup \{e_i\} \setminus \{f_i\} \in \mathcal{I}_1$
- $J \cup \{e_{i+1}\} \setminus \{f_i\} \in \mathcal{I}_2$

Since e_i, f_i share a vertex in V_1, then $J \cup \{e_i\} \setminus \{f_i\} \in \mathcal{I}_1$ makes sure no two edgs sharing a vertex in V_1 after symmetric difference; similar for e_{i+1}, f_i who share a vertex in V_2.
Let $J \in \mathcal{I}_1 \cap \mathcal{I}_2$. Suppose I find $P = \{e_1, f_1, \ldots, e_m, f_m, e_{m+1}\}$:

$$(\star)$$

- $e_i \notin J, \forall i = 1, \ldots, m+1$
- $f_i \in J, \forall i = 1, \ldots, m$

- $J \cup \{e_1\} \in \mathcal{I}_2$
- $J \cup \{e_{m+1}\} \in \mathcal{I}_1$
- $J \cup \{e_i\} \setminus \{f_i\} \in \mathcal{I}_1, \forall i = 1, \ldots, m$
- $J \cup \{e_{i+1}\} \setminus \{f_i\} \in \mathcal{I}_2, \forall 1, \ldots, m+1$

(That is, P is an augmenting path with respect to P), then define $J' = J \triangle P = J \cup \{e_1, \ldots, e_{m+1}\} \setminus \{f_1, \ldots, f_m\}$.

Lemma 54

If P is the smallest subset of S satisfying $(\star)$, then $J' \in \mathcal{I}_1 \cap \mathcal{I}_2$.

Proof.

Claim. $J \cup \{e_i\} \notin \mathcal{I}_1, \forall i = 1, \ldots, m$.

Proof. If $J \cup \{e_i\} \in \mathcal{I}_1$, pick $\{e_1, f_1, \ldots, e_{i-1}, f_{i-1}, e_i\}$ also satisfies $(\star)$ and smaller than P. □

Let $A = J \cup \{e_1, \ldots, e_{m+1}\}$, let $A_i = A \setminus \{f_m, \ldots, f_i\}$ (note $A_{m+1} = A$, $A_1 = J'$). Let C_i be the M_1-circuit in $J \cup \{e_i\}, \forall i = 1, \ldots, m$ (C_i exists by $J \cup \{e_i\} \notin \mathcal{I}_1$ from the claim).

Claim. $C_i \subseteq A_{i+1}$

Proof. Otherwise, by $C_i \subseteq J \cup \{e_i\} \subseteq A$, we have $C_i \nsubseteq A_{i+1}$ implies there exist $f_k \in C_i \subseteq A$ but not in A_{i+1}, so $f_k \in \{f_m, \ldots, f_{i+1}\}$. So there exist $k > i$ such that $J \cup \{e_i\} \setminus \{f_k\} \in \mathcal{I}_1$ because $J \cup \{e_i\}$ has a unique circuit C_i (by theorem 16), and $f_k \in C_i$. Then $\{e_1, f_1, \ldots, e_i, f_k, e_{k+1}, \ldots, f_m, e_{m+1}\}$ also satisfies $(\star)$, we find a smaller P, contradiction. □

Note

$$C_i \subseteq A_{i+1} = A_i \cup \{f_i\} \implies C_i \setminus \{f_i\} \subseteq A_i$$

Since C_i is an M_1-circuit in $J \cup \{e_i\}$, we know $C_i \setminus \{f_i\} \in \mathcal{I}_1$ by definition. Extend it to an M_1-basis of A_i, call it B_i. But $B_i \cup \{f_i\} \supseteq C_i$ and $B_i \subseteq A_i \subseteq A_{i+1} = A_i \cup \{f_i\}$. Hence, B_i is an M_1-basis of A_{i+1} which implies that $r_1(A_i) = r_1(A_{i+1}) \implies r_1(J') = r_1(A)$.
Then by $(\star)$, we know $J \cup \{e_{m+1}\} \in \mathcal{I}_1$, which implies

$$r_1(A) \geqslant |J|+1$$

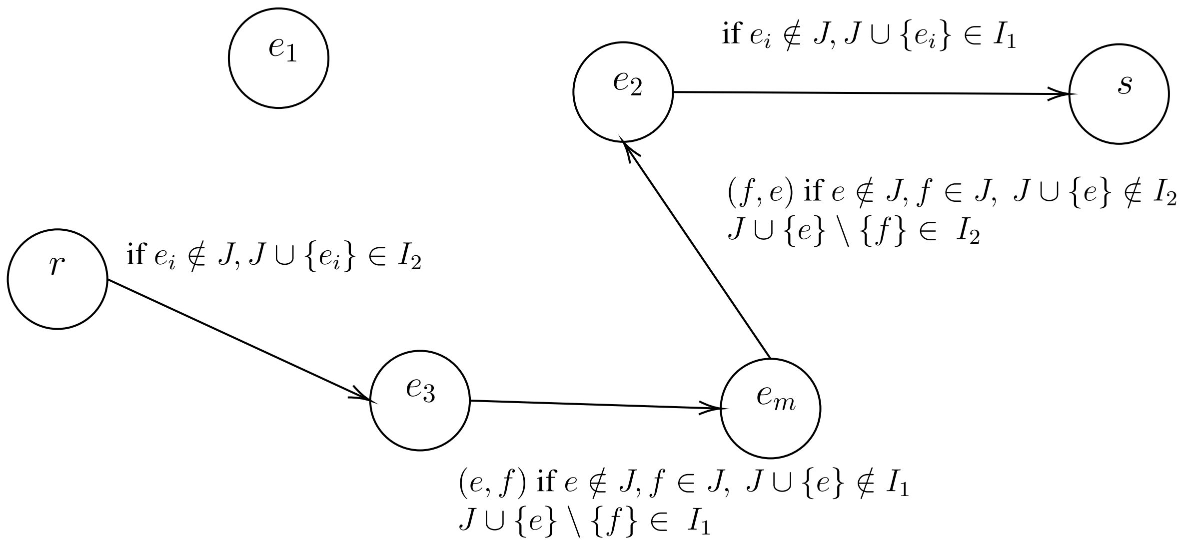
because $J \cup \{e_{m+1}\} \subseteq A$. And $|J|+1 = |J'|$ by definition. Hence

$$r_1(J') = r_1(A) \geqslant |J|+1 = |J'| \geqslant r_1(J')$$

that is, $|J'| = r_1(J')$, so $J' \in \mathcal{I}_1$. Similarly, we can prove $J' \in \mathcal{I}_2$, then we are done. □

5.2 Matroid Intersection Two

Recall that in last subsection, we consider the conditions $(\star)$, and we look for a P satsifying $(\star)$, but how to find P?
Finding P: $S = \{e_1, \ldots, e_m\}$, $J \in \mathcal{I}_1 \cap \mathcal{I}_2$.



According to the plot above, we treat each edge as a vertex, and add two more vertices r, s. If there is an edge $e_i \notin J, J \cup \{e_i\} \in \mathcal{I}_2$, then we add a directional edge from r to e_i. Similar for all other cases, then, if there is a directional path from r to s, by definition, the vertices on the directional path forms a P satisfying $(\star)$ in the original graph G.

Lemma 55

If there does not exist P satisfying $(\star)$, then J is a maximum cardinality element of $\mathcal{I}_1 \cap \mathcal{I}_2$.

Proof. Let U be elements of S reachable from r by directional path. If $U = \emptyset$, then for every $e_i \notin J$, $J \cup \{e_i\} \notin \mathcal{I}_2$. Then suppose there is a J' with larger cardinality, consider $e \in J' \setminus J$. Since $J \cup \{e\} \notin \mathcal{I}_2$, we know there exist $e' \in J$ such that e' and e share the same end in $\mathcal{I}_2$. That also implies that $e' \notin J'$. That is, for every $e \in J' \setminus J$, there exist a unique $e' \in J \setminus J'$, that is, $|J| \geqslant |J'|$, contradiction. Then by the definition of U, we know there exist an edge $e \in U \setminus J$.
Since (e, s) is not an arc (otherwise we find P), then $J \cup \{e\} \notin \mathcal{I}_1$. Let C be M_1-circuit of $J \cup \{e\}$. Since $\nexists (e, f)$ with $f \in S \setminus U$ (otherwise f is reachable from r, $f \in U$), we know $C \subseteq U$ (otherwise, $\exists f \in C \setminus U \subseteq J \cup \{e\} \setminus U = J \setminus U$, so $f \in S \setminus U$, and $J \cup \{e\} \setminus \{f\}$ has no circuit, so $J \cup \{e\} \setminus \{f\} \in \mathcal{I}_1$, so $\{e, f\}$ exists), so $C \subseteq (J \cap U) \cup \{e\}$. Also, $J \cap U \in \mathcal{I}_1$ implies that $J \cap U$ is an M_1-basis of U because $e \in U \setminus J$ is arbitrary, so for any $e \in U \setminus J$, $J \cap U \cup \{e\}$ contains a circuit. Then apply the same argument to show $J \cap \overline{U}$ is an M_2-basis of $\overline{U}$, basically, pick U^* be the elements of S that an reach s, then prove if U^* is empty, we are done, otherwise, $\overline{U} \supseteq U^*$ is not empty, similarly, we know $U^* \setminus J$ is not empty, so is $\overline{U} \setminus J$, then eveyrthing stays the same.Then

$$|J| = |J \cap U| + |J \cap \overline{U}| = r_1(U) + r_2(\overline{U})$$

implies J is optimal by Matroid Intersection Theorem. $\square$

A few notes:

- The lemma above leads to poyltime algorithm or solving matroid intersection (assuming checking independence can also be done in polytime).
- Weighted version also can be solved in polytime.
- Can also be solved using the LP:

$$\begin{aligned} max \quad & c^T x \\ \text{s.t.} \quad & x(A) \leqslant r_1(A), \forall A \subseteq S, \\ & x(A) \leqslant r_2(A), \forall A \subseteq S, \\ & \quad x \geqslant 0 \end{aligned}$$

- Solving $\max|A|$: $A \in \mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{I}_3$ is NP-hard (no known polynomial time algorithms for this, and most people believe there does not exist any).

5.2.1 Matroid Partitioning

Let $M_i = (S, \mathcal{I}_i)$ matroids for all $i = 1, \ldots, k$. Call $J \subseteq S$ partitionable if

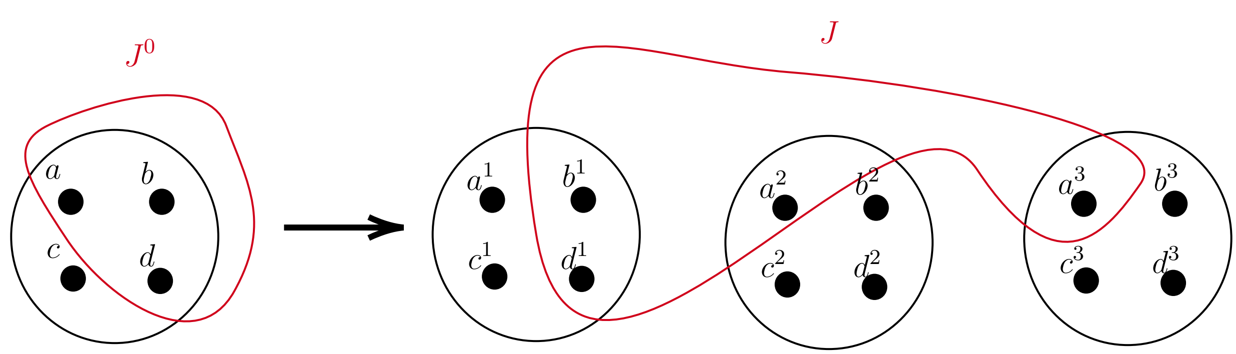
$$J = J_1 \dot\cup \ldots \dot\cup J_k, \text{ with } J_i \in \mathcal{I}_i, \forall i = 1, \ldots, k$$

Theorem 56: Matroid Partitioning; Edmonds&Fulkerson

$$\max\{|J| : J \text{ partitionable}\} = \min_{A \subseteq S} |\overline{A}| + \sum_{i=1}^{k} r_i(A)$$

Proof. Let $S = \{e_1, \ldots, e_n\}$. Let S^i be a copy of S (i.e. $S^i = \{e_1^i, \ldots, e_n^i\}$), for every $i = 1, \ldots, k$. For $J \subseteq \cup_{i=1}^k S^i$, let J^0 be corresponding set of elements in S:

$$J^0 = \{e \in S : \exists i \in \{1, \ldots, k\} \text{ such that } e^i \in J\}$$



Define $M'_i = (S^i, \{J \subseteq S^i : J^0 \in \mathcal{I}_i\})$. Let $N_a = M'_1 \oplus \ldots \oplus M'_k$. Let $S' = \cup_{i=1}^k S_i$, $\mathcal{I}_b = \{A \subseteq S' : A$ has at most one copy of $e, \forall e \in S\}$, $N_b = (S', \mathcal{I}_b)$.
Suppose there exist $J \in \mathcal{I}_a \cap \mathcal{I}_b$. Then J^0 is partitionable and $|J^0|= |J|$. The reason is, since $J^0 \in \mathcal{I}_a$, we partition J^0 with respect to each M'_i; since $J^0 \in \mathcal{I}_b$, for every $e^i \in J$, $e \in J^0$ and there exists at most one i such that $e^i \in J$, so $|J^0| \geqslant |J|$, and by definition $|J^0| \leqslant |J|$, so $|J^0|= |J|$. Thus

$$\max\{|J|: J \in \mathcal{I}_a \cap \mathcal{I}_b\} \leqslant \max\{|J^0|: J^0 \text{ is partitionable}\}$$

Conversely, suppose some set K is partitionable, then we can put each partition of it into M'_i, then we create $K' \in \mathcal{I}_a \cap \mathcal{I}_b$ such that $|K'|= |K|$ and $K = (K')^0$. Thus,

$$\max\{|J|: J \in \mathcal{I}_a \cap \mathcal{I}_b\} \geqslant \max\{|J^0|: J^0 \text{ is partitionable}\}$$

Then by Matroid Intersection Theorem,

$$\max\{|J^0|: J^0 \text{ is partitionable}\} = \min_{B \subseteq S'}\{r_a(B) + r_b(S' \setminus B)\}$$

We may assume the minimizer B is in the form: $\cup_{e \in B^0}\{e^1, \ldots, e^k\}$. The reason is, suppose there exists $e^j \in B$, and $e^k \in S' \setminus B$. Consider $B' = B \setminus \{e^j\}$. Let D be N_b-basis of $S' \setminus B$, and note $D \subseteq S' \setminus B'$. If $D \cup \{e^j\} \in \mathcal{I}_b$, then $e^k \notin D$ (otherwise $D \cup \{e^j\}$ has two copies of e), so $D \cup \{e^k\} \in \mathcal{I}_b$, D is not an N_b-basis of $S' \setminus B$, contradiction. Hence, D is an M_b basis of $S' \setminus B'$. So $r_b(S' \setminus B') = r_b(S' \setminus B)$. Moreover, $r_a(B') \leqslant r_a(B)$, so B' is also a minimizer.
Then, we have $r_a(B) = \sum_{i=1}^k r_i(B^0)$ and $r_b(S' \setminus B) = |S \setminus B^0|$, we are done. $\square$

6 T-Join

6.1 T-Join One

Definition 57: Euler Tours

Given a connected graph $G = (V, E)$ (potentially having parallel edges), an Euler Tour is a closed walk visiting every edge (not necessary every vertex) exactly once.

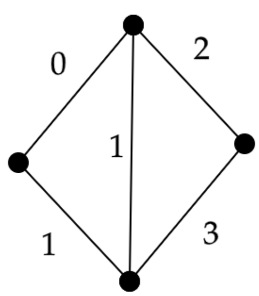
Theorem 58

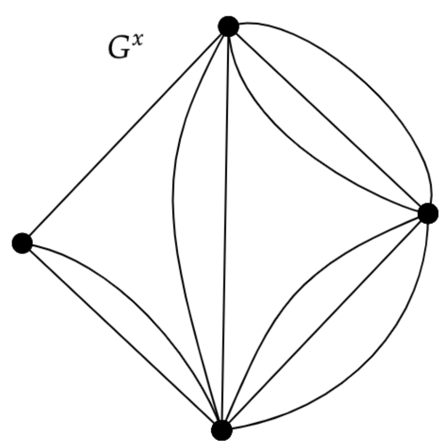
A connected graph G has an Euler tour if and only if every vertex has even degree.

Definition 59: Postman Tour

A postman tour is a closed walk traversing every edge at least (not necessary to be exactly) once.

Consider the situation that $c_e \geqslant 0, \forall e \in E$, every time e is traversed, it incurs cost c_e.
Goal: Find minimum cost postman tour.
Note if G has an Euler tour, it is optimal.
Let $x_e \in Z, x_e \geqslant 0, \forall e \in E$. Let G^x be obtained by making $1 + x_e$ copies of e (all with cost c_e).





<u>Idea:</u>Find x so that G^x has Euler tour.

$$\begin{aligned} min \quad & \sum_{e\in E} c_e x_e \\ \text{s.t.} \quad & \sum_{e\in\delta(v)} (1+x_e) \equiv 0 \mod 2, \forall v \in V, \\ & x_e \geqslant 0, \\ & x \in Z^E \end{aligned}$$

which is equivalent to

$$\begin{aligned} min \quad & \sum_{e\in E} c_e x_e \\ \text{s.t.} \quad & x(\delta(v)) \equiv |\delta(v)| \mod 2, \forall v \in V, \\ & x_e \geqslant 0, \\ & x \in Z^E \end{aligned}$$

Note. Since $c_e \geqslant 0$, we may assume $x_e \in \{0,1\}, \forall e \in E$. If $x_e \geqslant 2$, let $x'_e \leftarrow x_e - 2$, then the cost decrease by $2c_e$, but the constraints are still satisfied.

So the problem becomes

$$\begin{aligned} min \quad & \sum_{e\in E} c_e x_e \\ \text{s.t.} \quad & x(\delta(v)) \equiv |\delta(v)| \mod 2, \forall v \in V, \\ & x_e \in \{0,1\} \qquad ,\forall e \in E \end{aligned}$$

Want: (Postman set) Set $J \subseteq E$ such that $|J \cap \delta(v)| \equiv |\delta(v)| \mod 2$.

Definition 60: T-joins

More generally, let $T \subseteq V$ such that $|T|$ is even. $J \subseteq E$ is called a <u>T-join</u> if

$$|J \cap \delta(v)| \equiv |T \cap \{v\}| \mod 2, \forall v \in V$$

that is, the vertices of (V, J) with odd degree are precisely T. Also, every vertex in $V \setminus T$ has even degree in (V, J). That is, you can say $v \in T$ if and only if $|J \cap \delta(v)|$ is odd.

6.1.1 Min Cost T-Join

Given $c \in \mathbb{R}^E, G = (V, E), T \subseteq V$ with $|T|$ even, find a T-join of G minimizing $c(J)$.

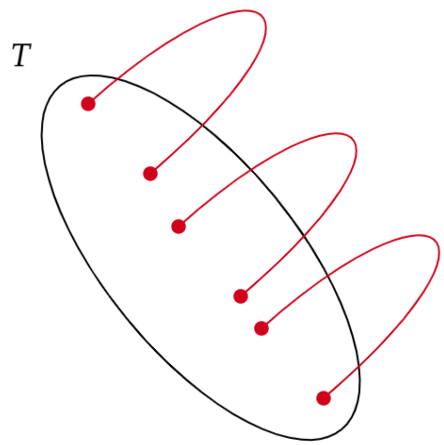
Note. Definition does not require connectedness or $c \geqslant 0$.

Example 61

- Postman sets.
- $T = \emptyset \implies$ every vertex has even degree in (V, J), finding negative cost cycles.
- $T = \{r, s\}$. Only r, s have odd degree in (V, J), so the solution will be r-s path plus cycles.

6.1.2 Min Cost T-Joint when $c \geqslant 0$

There is always an optimal T-join that is minimal (inclusionwise), so we can focus on minimal T-joins. But what does minimal T-joins look like?



Proposition 62

J is a minimal T-join if and only if it is the union of the edges of $\frac{|T|}{2}$ edge-disjoint paths, joining pairs of vertices in T (all distinct).

Proof. Let $|T|= 2k$, $k \in Z_+$. Induction on k.
Base case: $k = 0$, $T = \emptyset$, the only minimal T-join is $\emptyset$. Let $|T|= 2k, k \in Z_+$.

- ($\impliedby$) Trivial. First, it is clear that J is a T-join. If it is not minimal, then we know $J' = J \setminus \{e\}$ is a T-join, but then the two ends of e are in T, contradiction.
- ($\implies$) Suffices to show J contains such edge set (because if there is an egde in J but not in this set, then J is not minimal since by above, the edge set is always a minimal T-join).
  Let $u \in T$, K be the connected component of (V, J) containing u. There must exists $v \in T \setminus u$ such that $v \in K$, since only vertices in T have odd degree in (V, J). That is, if v not exists, then K only has one odd degree vertex in (V, J), then the sum of degree of K is odd, contradiction. Let P be u-v path in (V, J). Consider $J' = J \setminus E(P)$. If we show J' is a

T'-join ($T' = T \setminus \{u, v\}$), then by induction we are done. It is not hard to show, for every $x \in V$, if $x \notin P$, then the number of degree does not change, so $|J' \cap \delta(v)| = |J \cap \delta(v)|$ which is still odd/even; if $x \in P$, then if $x \notin \{u, v\}$ then $|J' \cap \delta(v)| = |J \cap \delta(v)| - 2$ which is still odd; if $x \in \{u, v\}$, then it has even degree in (V, J'). Hence, J' is a T'-join, and it is minimal, otherwise, we can delete an edge e from J' and it is still a T'-join, then if we delete e from J, it will still be a T-join, contradiction.

□

Proposition 63

Let J' be a T'-join of G. Then J is a T-join of G if and only if $J \triangle J'$ is a $(T \triangle T')$-join of G.

Proof.

- ($\Longrightarrow$) Let $\overline{J} = J \triangle J'$. Let $v \in V$. If $v \in T, v \notin T'$ then $v \in T \triangle T'$. We have $|J \cap \delta(v)|$ being odd and $|J' \cap \delta(v)|$ being even. Then $|J \cap \delta(v)| + |J' \cap \delta(v)|$ is odd. But $J \triangle J'$ removes even number of those, that is

$$|\overline{J} \cap \delta(v)| = |J \cap \delta(v)| + |J' \cap \delta(v)| - 2|J \cap J' \cap \delta(v)|$$

  so $|\overline{J} \cap \delta(v)|$ is odd. Similar argument applies to other cases. Hence,

$$|\overline{J} \cap \delta(v)| \text{being odd} \iff v \in T \triangle T'$$

- ($\Longleftarrow$) Apply forward implication: with $J' = J', J'' = J \triangle J'$, and $T' = T', T'' = T \triangle T'$, then by the forward direction

$$(J \triangle J' \triangle J', T \triangle T' \triangle T') = (J \triangle (J' \triangle J'), T \triangle (T' \triangle T')) = (J, T)$$

  implies J is a T-join.

□

Proposition 64

Suppose $c \geqslant 0$, then there exists a minimum cost T-join that is the union of $|T|/2$ edge-disjoint shortest paths joining vertices of T in pairs (all distinct). Here the shortest paths means the paths with least costs.

Proof. Let J be a minimal minimum cost T-join. Let P be a u-v path with $E(P) \subseteq J, u, v \in T$. Suppose P' is a u-v path with $c(E(P')) < c(E(P))$. Note $E(P), E(P')$ are $\{u, v\}$-joins. Then $J' = J \triangle E(P) \triangle E(P')$ is a T-join (actually a $T \triangle \{u, v\} \triangle \{u, v\}$-join). Then

$$\begin{aligned} c(J') &= c(J \setminus E(P)) + c(E(P')) - 2c((J \setminus E(P)) \cap E(P')) \\ &\leqslant c(J) - c(E(P)) + c(E(P')) \\ &< c(J) \end{aligned}$$

contradiction. □

Can you propose an algorithm to solve min. cost T-join when $c \geqslant 0$?
Let $d(u, v)$ be the cost of shortest u-v path. Construct G' with cost $d(u, v)$ for every edge in G', a complete graph with vertex set T.

Claim. Min cost T-join when $c \geqslant 0$ can be computed by computed min weight perfect matching in G' (note the weight of edge u, v is $d(u, v)$ and G' has even vertices and complete).

Let M be a min weighted perfect matching in G'. Now let $\{u_i, v_i\}_{i=1}^{|T|/2}$ be the edges in M. Let P_i corresponding shortest paths in G, $\forall i = 1, \dots, \frac{|T|}{2}$. Then $E(P_1)\triangle \dots \triangle E(P_{|T|/2})$ is a T-join of cost $\leqslant \sum_{i=1}^{|T|/2} d(u_i, v_i)$. By proposition, any T-join that is minimal corresponds to a matching in G' and has cost at least $\sum_{i=1}^{|T|/2} d(u_i, v_i)$. Hence, $E(P_1)\triangle \dots \triangle E(P_2)$ is a min cost T-join. Note that we keep use minimal T-join because by what we observe previously, when $c \geqslant 0$, there exists an optimal T-join that is minimal.

6.2 T-Joint Two

6.2.1 Min-Cost T-join for arbitrary costs

Let $N = \{e \in E : c_e < 0\}$. Let $T' = \{v \in V : v \text{ has odd degree in } (V, N)\}$. Then N is a T'-join. Note by proposition 63 above, we know

$$J \text{ is a } T\text{-join} \iff J\triangle N \text{ is a } (T\triangle T')\text{-join}$$

$$\begin{aligned} c(J) &= c(J \setminus N) + c(J \cap N) \\ &= c(J \setminus N) - \underbrace{c(N \setminus J)}_{c(e) \leqslant 0, \forall e \in N \setminus J} + \underbrace{c(N \setminus J) + c(J \cap N)}_{c(N)} \\ &= \sum_{e \in J \triangle N} |c_e| + c(N) \end{aligned}$$

To minimize $c(J)$, suffices to find min cost $(T\triangle T')$-join with respect to costs $|c_e| \geqslant 0, \forall e \in E$ because $c(N)$ is a constant.
Algorithm:

- Find min cost $(T\triangle T')$-join with respect to costs $|c_e|, \forall e \in E$ (call it J^*).
- Output min costs T-join $J^*\triangle N$.

6.2.2 LP formulations

Definition 65

A set $S \subseteq V$ is T-odd if $|S \cap T|$ is odd. If S is T-odd, $\delta(S)$ is a T-cut.

Let $S \subseteq V$ be T-odd; J be a T-join. If $J \cap \delta(S) = \emptyset$, then the subgraph of (V, J) induced by S

has an odd number of odd degree vertices (because every vertex in $S \cap T$ has odd degree in (V, J), and every edge in J incident to v is uv for a $u \in S$.), contradiction. Hence, $|J \cap \delta(S)| \geqslant 1$.

$$\begin{aligned} (P)\ min \quad & \sum_{e \in E} c_e x_e \\ \text{s.t.} \quad & x(\delta(S)) \geqslant 1, \forall T\text{-odd set } S, \\ & x \geqslant 0 \end{aligned}$$

Theorem 66

Let $G = (V, E), T \subseteq V$, $|T|$ even, $c \in \mathbb{R}^E, \geqslant 0$. Then the min cost of a T-join is equal to the optimal value of (P).

Proof. Let J^* be an optimal T-join. We've seen that the indicator vector of J^* is feasible for (P), so $c(J^*) \geqslant \zeta^*$ (ζ^* is the optimal value of (P)). Let ϑ be the set of T-odd sets. Then

$$\begin{aligned} (P)\ min \quad & \sum_{e \in E} c_e x_e \\ \text{s.t.} \quad & x(\delta(S)) \geqslant 1, \forall S \in \vartheta, \\ & x \geqslant 0 \end{aligned}$$

$$\begin{aligned} (D)\ max \quad & \sum_{S \in \vartheta} \alpha_S \\ \text{s.t.} \quad & \sum_{S \in \vartheta : e \in \delta(S)} \alpha_S \leqslant c_e, \forall e \in E, \\ & \alpha \geqslant 0 \end{aligned}$$

<u>Case1:</u> $T = V$. Let $G' = (V, E')$ be graph used to solve min cost T-join with costs d (min costs from u to v). Then we have the min weighted perfect matching LP:

$$\begin{aligned} (P_M)\ min \quad & \sum_{u,v \in E'} d(u, v) w_{uv} \\ \text{s.t.} \quad & w(\delta(v)) = 1, \forall v \in V, \\ & w(\delta(A)) \geqslant 1, \forall A \subseteq V : |A| \geqslant 3, |A| \text{ odd}, \\ & w \geqslant 0 \end{aligned}$$

Let $\vartheta_M = \{A \subseteq V : |A| \geqslant 3, |A| \text{ odd}\}$. Then

$$\begin{aligned} (D_M)\ max \quad & \sum_{v \in V} \beta_v + \sum_{A \in \vartheta_M} \gamma_A \\ \text{s.t.} \quad & \beta_u + \beta_v + \sum_{A \in \vartheta_M : uv \in \delta(A)} \gamma_A \leqslant d(u, v), \forall u, v \in E', \\ & \gamma_A \geqslant 0 \end{aligned}$$

We've shown that $c(J^*)$ is the optimal value to (P_M) and (D_M).
Idea: Note $E \subseteq E'$, $d(u,v) \leqslant c_{uv}, \forall u,v \in E$. From optimal solution to (D_M), build optimal solution to (D) of same cost. Then get α such that $c(\alpha) = c(J^*) \geqslant \zeta^*$.
Gneral T: Build $\hat{G}$ with a copy $\hat{v}$ of v for all $v \in V \setminus T$ and add $v\hat{v}$ edges, with cost 0. Let $\hat{T} = \hat{V}$ and find min cost $\hat{T}$-join of $\hat{G}$, call it $\hat{J}$.
Note that, since every copied vertex $\hat{v}$ has degree one in $\hat{G}$, and $\hat{v} \in \hat{T}$, so it has degree one in $(\hat{V}, \hat{J})$, that is, $v\hat{v}$ is in $\hat{J}$. Let J be the set deleting all such $v\hat{v}$. It is a T-join because, for every $v \in T$, $\delta(v) \cap T = \delta(v) \cap T'$, so it has odd degree in (V,T). for every $v \notin T$, $|\delta(v) \cap T| = |\delta(v) \cap T'| - 1$ which is even. Hence, J is a T-join.
Idea: From dual solution to $\hat{G}$, construct dual solution to G. $\square$

7 Flows and Cuts

7.1 Flows and Cuts One

Definition 67

Consider a directed graph $D = (V, A)$ (where A represents arcs), and $x \in \mathbb{R}^A$. Let $r, s \in V$. We say x is an $\underline{r\text{-}s \text{ flow}}$ if

$$x(\delta^-(v)) - x(\delta^+(v)) = 0, \forall v \in V \setminus \{r, s\}$$

where

$$\delta^-(S) = \{(u, v) \in A : u \notin S, v \in S\}$$
$$\delta^+(S) = \{(u, v) \in A : u \in S, v \notin S\}$$

and we call

$$f_x(v) = x(\delta^-(v)) - x(\delta^+(v))$$

the **Net flow into** v.

Definition 68

Given capacities $l \leqslant u$, $l, u \in \mathbb{R}^A$ (over the arc set A), an r-s flow x is feasible if

$$l_a \leqslant x_a \leqslant u_a, \forall a \in A$$

Given a feasible r-s flow x, its value is $f_x(s)$ (the flow entering s).

Max-Flow Problem:
Find feasible r-s flow of maximum value.
Max-Flow Problem:(integer)
Find feasible integer r-s flow of maximum value.

We will assume $l = 0$, $\delta^-(r) = \delta^+(s) = \emptyset$.

Definition 69

For $R \subseteq V$, we call $\delta^+(R)$ a (directed) cut. Moreover if $r \in R, s \notin R$, we say $\delta^+(R)$ is an $\underline{r\text{-}s \text{ cut}}$.

Proposition 70

If x is a feasible r-s flow and $\delta^+(R)$ is an r-s cut, then

$$x(\delta^+(R)) - x(\delta^-(R)) = f_x(s)$$

Proof. x being a feasible r-s flow implies that $f_x(v) = 0, \forall v \in V \setminus \{r, s\}$. Proof can be done by adding $f_x(r) = f_x(s) = 0$. □

Corollary 71

If x is a feasible r-s flow, $\delta^+(R)$ being an r-s cut implies $f_x(s) \leqslant u(\delta^+(R))$.

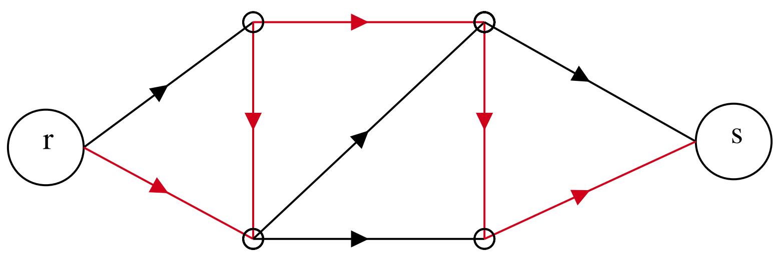
Proof.

$$f_x(s) = x(\delta^+(R)) - \underbrace{x(\delta^-(R))}_{\geqslant 0} \leqslant x(\delta^+(R)) \leqslant u(\delta^+(R))$$

□

Definition 72

Suppose P is a path that uses some arcs in "forward" direction, some in "backward" direction. E.g.



We say that P is <u>x-incrementing</u> if

- $x_a < u_a$, for every a appearing in forward direction.
- $x_a > 0$, for every a appearing in backward direction.

We say P is x-augmenting if it is an x-incrementing r-s path.

Note. If there exists an x-augmenting path, then x is NOT a max flow, because we can

- For every forward arc, increase x_a by ϵ.
- For every backward arc, decreas x_a by ϵ.

Hence, we increase $f_x(s)$ by ϵ.
Let's say we find x and it has an x-augmenting path, and by the operations above, we get x^*. Then, for every vertex except r, s on the path, if it's incident to two forward or backward arcs, then one is in and the other is out, so net into flow is still 0; if it's incident to one forward and one backward. Either both of them are out or both of them are in arcs for the vertex, so the total net into flow still does not change, so x^* is feasible.

Theorem 73: Max Flow/Min Cut

If there exists a max flow, then

$$\max\{f_x(s) : x \text{ is a feasible } r\text{-}s \text{ flow}\} = \min\{u(\delta^+(R)) : \delta^+(R) \text{ is an } r\text{-}s \text{ cut}\}$$

Proof. Let x be a max-flow. Let $R = \{v \in V : \exists \text{ an } r\text{-}v\ x\text{-incrementing path}\}$.
Note: $r \in R, s \notin R \implies \delta^+(R)$ is an r-s cut. (if $s \in R$, then there is an x-augmenting path, x is not a max flow)
Also, let $(v,w) \in \delta^+(R)$, then $x_{vw} = u_{vw}$; otherwise $w \in R$ because r-vw is an r-w incrementing path.
Similarly, let $(v,w) \in \delta^-(R)$, then $x_{vw} = 0$, becasue otherwise $v \in R$. Hence,

$$f_x(s) = x(\delta^+(R)) - \underbrace{x(\delta^-(R))}_{=0} = u(\delta^+(R))$$

and we know $u(\delta^+(R))$ is the minimum of the RHS by Corollary 71. □

Note. This can also be shown via LP:

$$\begin{aligned} max \quad & f_x(s) \\ \text{s.t.} \quad & x(\delta^-(v)) - x(\delta^+(v)) = 0 \ , \quad \forall v \in V \setminus \{r,s\}, \\ & \qquad\qquad 0 \leqslant x_a \leqslant u_a, \quad \forall x \in A \end{aligned}$$

$$\begin{aligned} min \quad & \sum_{a \in A} u_a \zeta_a \\ \text{s.t.} \quad & -y_v + y_w + \zeta_{vw} \geqslant 0, && \forall (v,w) \in A : v,w \in V \setminus \{r,s\}, \\ & -y_r + y_w + \zeta_{rw} \geqslant 0, && \forall (r,w) \in A, \\ & -y_v + \zeta_{vs} \geqslant 1, && \forall (v,s) \in A, \\ & \zeta_{rs} \geqslant 1, && \text{if } (r,s) \in A, \\ & \zeta \geqslant 0 \end{aligned}$$

which can be written as

$$\begin{aligned} min \quad & \sum_{a \in A} u_a \zeta_a \\ \text{s.t.} \quad & -y_v + y_w + \zeta_{vw} \geqslant 0 \ , && \forall (v,w) \in A : v,w \in V \setminus \{r,s\}, \\ & -y_r + y_w + \zeta_{rw} \geqslant 0 \ , && \forall (r,w) \in A, \\ & -y_v + y_s + \zeta_{vs} \geqslant 0 \ , && \forall (v,s) \in A, \\ & -y_r + y_s + \zeta_{rs} \geqslant 0 \ , && \text{if } (r,s) \in A, \\ & \zeta \geqslant 0, \\ & y_r = 0, \\ & y_s = -1 \end{aligned}$$

we can simplify the dual problem as

$$\begin{aligned}
min \quad & \sum_{a\in A} u_a\zeta_a \\
\text{s.t.} \quad & -y_v + y_w + \zeta_{vw} \geqslant 0 \quad , \quad \forall (v,w)\in A, \\
& \zeta \geqslant 0, \\
& y_r = 0, \\
& y_s = -1
\end{aligned}$$

and we can also add constants to all y because it doesn't violate the constraints nor changing the objective value, so

$$\begin{aligned}
min \quad & \sum_{a\in A} u_a\zeta_a \\
\text{s.t.} \quad & -y_v + y_w + \zeta_{vw} \geqslant 0, \quad \forall (v,w)\in A, \\
& \zeta \geqslant 0, \\
& y_r = 1, \\
& y_s = 0
\end{aligned}$$

Then we can argue that it has optimal solution with $y_v = 1$ if $v \in R$; $y_v = 0$ otherwise AND $\zeta_{vw} = 1$ if $vw \in \delta^+(R)$; $\zeta_{vw} = 0$ otherwise.

Theorem 74

A feasible r-s flow x is maximum if and only if there does not exist an x-augmenting path.

Theorem 75

If $u \in Z_+^A$ and there exists a max flow, then there exists a max flow that is integral.

Proof. First, it is clear that we can find a max integral flow (that is, an integral flow which is the max among all other integral flow). Let x be a max. integral flow, and assume it is not a max flow. Then, we know there is an x-augmenting path, however, since, $u \in \mathbb{Z}_+^A$, we know the ϵ we can pick to add/subtract on forward/backward arcs is also an integer, that is, x is not a maximum integral flow, contradiction. □

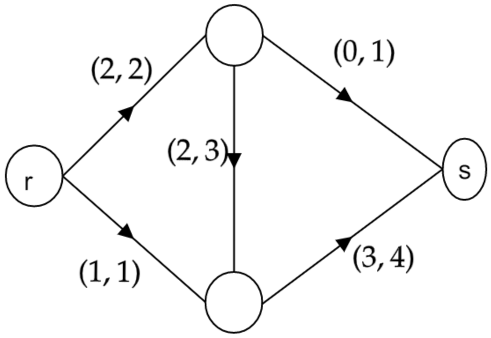
7.1.1 Ford-Fulkerson Algorithm:

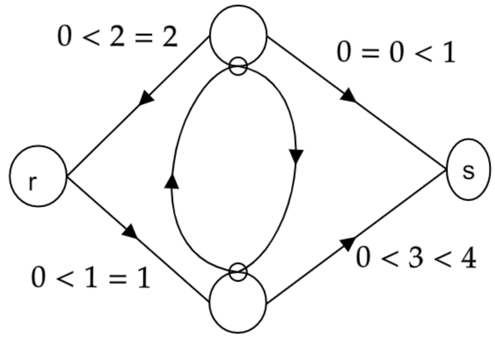
Idea:Construct $D_x = (V, A_x)$ where

- $vw \in A_x$ if $vw \in A$ and $x_{vw} < u_{vw}$
- $vw \in A_x$ if $wv \in A$ and $x_{wv} > 0$

that is, D_x consists of arcs that can appear in x-incrementing/x-augmenting paths.

Example 76



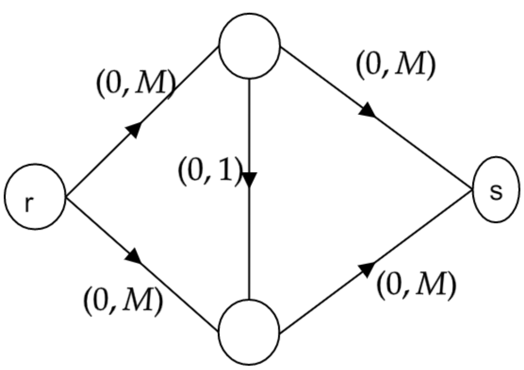


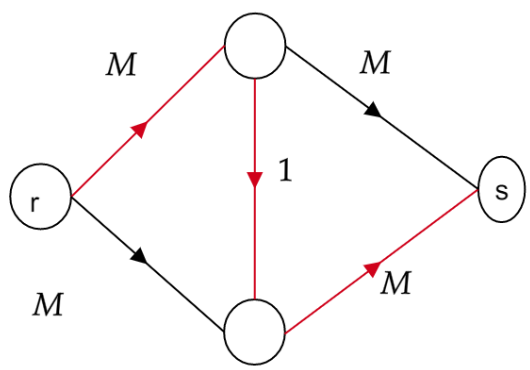
- Let $u_{vw} - x_{vw}$ be residual capacity of forward arcs vw in D_x.
- Let x_{vw} be residual capacity of backward arcs wv in D_x

where in D_x, if the arc has the same direction as in G, we call it forward, otherwise we call it backward.

There exists x-augmenting path in G if and only if there exists r-s directed path in D_x.

Moreover, if P is such a path, $f_x(s)$ can be increased by smallest residual capacity in P. Also, an augmenting path can be found in $O(m)$ time ($m = |A|$). But how many times do we need to do the augmenting steps to find the max flow?





If we do the augmenting above, and with some choices of the path, we may need $2M$ iterations to terminate.

7.2 Flows and Cuts Two

Theorem 77: Dinits 70;Edmonds& Karp 72

If P is chosen to be the shortest r-s path in D_x (shortest w.r.t. the number of arcs), then there are $\leqslant nm$ augmentations.

Proof. Let $d_x(v,w)$ be the length of shortest v-w path in D_x. Let $P = v_0,\dots,v_k$ be shortest r-s path in D_x and x' be feasible r-s flow obtained after augmenting x using P.

Claim. $\forall v \in V$, $d_{x'}(r,v) \geqslant d_x(r,v)$ and $d_{x'}(v,s) \geqslant d_x(v,s)$, note the distance might by infinity if no such path.

Proof. Suppose there exists $v : d_{x'}(r,v) < d_x(r,v)$. Choose such v with the smallest $d_{x'}(r,v)$. Let P' be r-v path of $D_{x'}$ with length $d_{x'}(r,v)$. Let w be vertex immediately before v in P'. Then

$$d_x(r,v) > d_{x'}(r,v) = d_{x'}(r,w) + 1 \geqslant d_x(r,w) + 1 \dots (*)$$

$d_{x'}(r,w) \geqslant d_x(r,w)$ otherwise we have a contradiction to the choice of v.
If $wv \in A_x$, then $d_x(r,v) \leqslant d_x(r,w) + 1$ because we can go from r to w first then v in D_x; if v is on the shortest path from r to w, then $d_x(r,v) \leqslant d_x(r,w)$; so we reach a contradiction.
If $wv \notin A_x$, $wv \in A_{x'}$ implies that the residual of wv or vw in G is changed which implies wv or vw is an arc in P. But $E(P) \subseteq A_x \implies vw \in E(P)$ which implies $v = v_{i-1}, w = v_i$ for some $i = 1,\dots,k$. Then combine with $(*)$, we have

$$d_x(r, v_{i-1}) \geqslant d_x(r, v_i) + 1$$

since P is the shortest path from r to s in D_x, if the above is true, we can go to v_i first and avoid v_{i-1} in P, contradiction. □

This claim shows that the algorithm works in at most $n-1$ stages (in each stage $d_x(r,s)$ remains constant).

Claim. If $d_{x'}(r,s) = d_x(r,s)$, then $\tilde{A}_{x'} \subsetneq \tilde{A}_x$,
where $\tilde{A}_x := \{vw \in A :$ either vw or wv are in a shortest x-augmenting path$\}$.

Proof. Let $k = d_x(r,s), vw \in \tilde{A}_{x'}$. If vw in shortest r-s path $D_{x'}$, then there exists i:

$$d_{x'}(r,v) = i-1, d_{x'}(w,s) = k-i \implies d_{x'}(r,v) + d_{x'}(w,s) = k-1$$

By previous claim,

$$d_x(r,v) + d_x(w,s) \leqslant k-1$$

If $vw \notin \tilde{A}_x$, then we know it is not in the shortest x-augmenting path of A_x, so its flow doesn't change; then $x_{vw} = x'_{vw} \implies vw \in A_x$ because $vw \in A_{x'}$ and the flow doesn't change. But then there exists r-s path of length at most k (the path $r - v - w - s$), so $vw \in \tilde{A}_x$. Similar in the case wv is in shortest r-s path in $D_{x'}$. Hence $\tilde{A}_{x'} \subseteq \tilde{A}_x$.
Now let P be the path used to change x to x'. There exists $vw \in A$:

- $vw \in P$ and $x'_{vw} = u_{vw}$ or

- $wv \in P$ and $x'_{vw} = 0$

For the first case, $d_x(r,v) = i-1$, $d_x(w,s) = k-i$, $vw \in \tilde{A}_x$ and $vw \notin D_{x'}$ because $x'_{vw} = u_{vw}$. An x'-augmenting path cannot use vw implies that if $vw \in \tilde{A}_x$, then there exists an x'-augmenting path using wv. But $d_{x'}(r,w) + d_{x'}(v,s) \geqslant d_x(r,w) + d_x(v,s) = (i-1+1)+(k-i+1) = k+1$ by previous claim, so any x'-augmenting path using wv has length at least $k+2$, so $vw \notin \tilde{A}_{x'}$. Similar for the other case. Hence, $\tilde{A}_{x'} \subsetneq \tilde{A}_x$. □

The first claim shows algorithm works in at most $n-1$ stages and the second one shows each stage has at most m iterations. □

7.2.1 Applications

- Bipartite matching/König's Theorem
- Assignment problems
- Flow feasibility
- ...

For the bipartite matching/König's Theorem, we add r and s to the graph, let (A,B) be the bipartition. Add arcs (r,v) for every $v \in A$ with capacity 1; add arcs (v,s) for every $s \in B$ with capacity 1; and add arcs (v,u) for every $vu \in E(G), v \in A, u \in B$ with capacity 1.

For the flow feasibility problem: let $D = (V,A)$, $u \in \mathbb{R}^A_+, b \in \mathbb{R}^V$ such that $b(V) = 0$, does there exists $x \in \mathbb{R}^A : f_x(v) = b_v, \forall v \in V$ and $0 \leqslant x_a \leqslant u_a, \forall a \in A$?
Formally, add r, s to V, add arcs (r,v) with capacity $-b_v$ if $b_v < 0$, and (v,s) with capacity b_v if $b_v > 0$ and compute max flow.

Such a flow we want exists if and only if the max r-s flow is $\sum_{v:b_v>0} b_v$

which is equivalent to

$$\begin{aligned}
&\iff \forall S \subseteq V, u(\delta^+(S \cup \{r\})) \geqslant \sum_{v:b_v>0} b_v \\
&\iff \sum_{v \notin S: b_v>0} b_v + \sum_{v \notin S: b_v<0} b_v \leqslant u(\delta^+_D(s)) \\
&\iff \forall S \subseteq V, b(s) \leqslant u(\delta^+_D(\overline{S}))
\end{aligned}$$

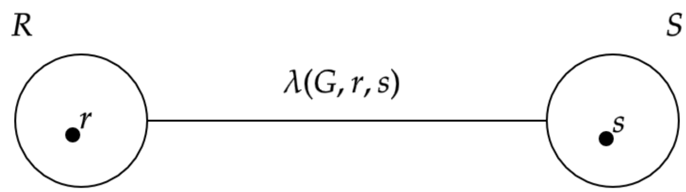
where from the first line to the second, we use $u(\delta^+(S \cup \{r\})) = \sum_{v \in S: b_v>0} b_v = \sum_{v \notin S: b_v>0}(-b_v) + u(\delta^+_D(S))$; from the second line to the last line, we use a theorem by Gale (57).

7.3 Flows and Cuts Three

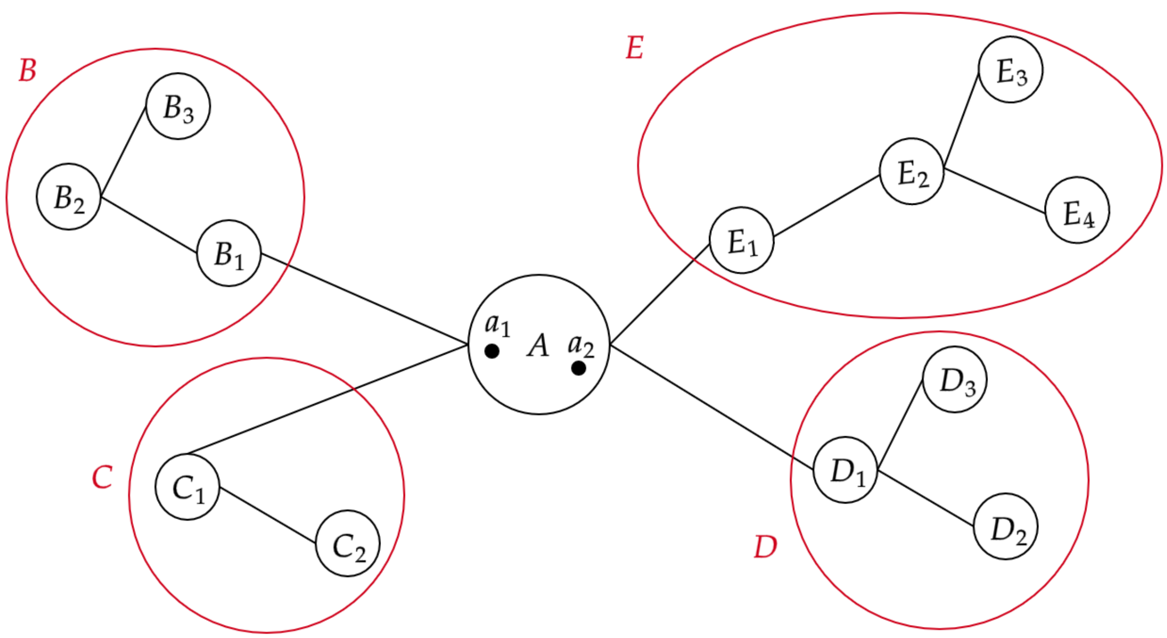
Undirected MinCut
Given $G = (V,E)$ undirected, $u \in \mathbb{R}^E_+$, find $S \subseteq V : \emptyset \subsetneq S \subsetneq V$ minimizing $u(\delta(S))$.
Let $\lambda(G)$ be the weight of min cut of G. $\lambda(G,v,w)$ be the weight of min v-w cut (i.e. $v \in S, w \notin S$).
Idea: Direct graph, compute max v-w for every v, w, which takes $O(n^2)$ max flow computations.

7.3.1 Gomory-Hu Trees

Pick $r, s \in V$ arbitrary and compute min r-s cut. Let R, S be its "shores", i.e. $\lambda(G, r, s) = u(\delta(R)), S = V \setminus R$. Treat R and S as two vertices, we create a tree:



where let $\lambda(G, r, s)$ be the edge label between R and S.
In general, suppose our current Gomory-Hu Tree T is:

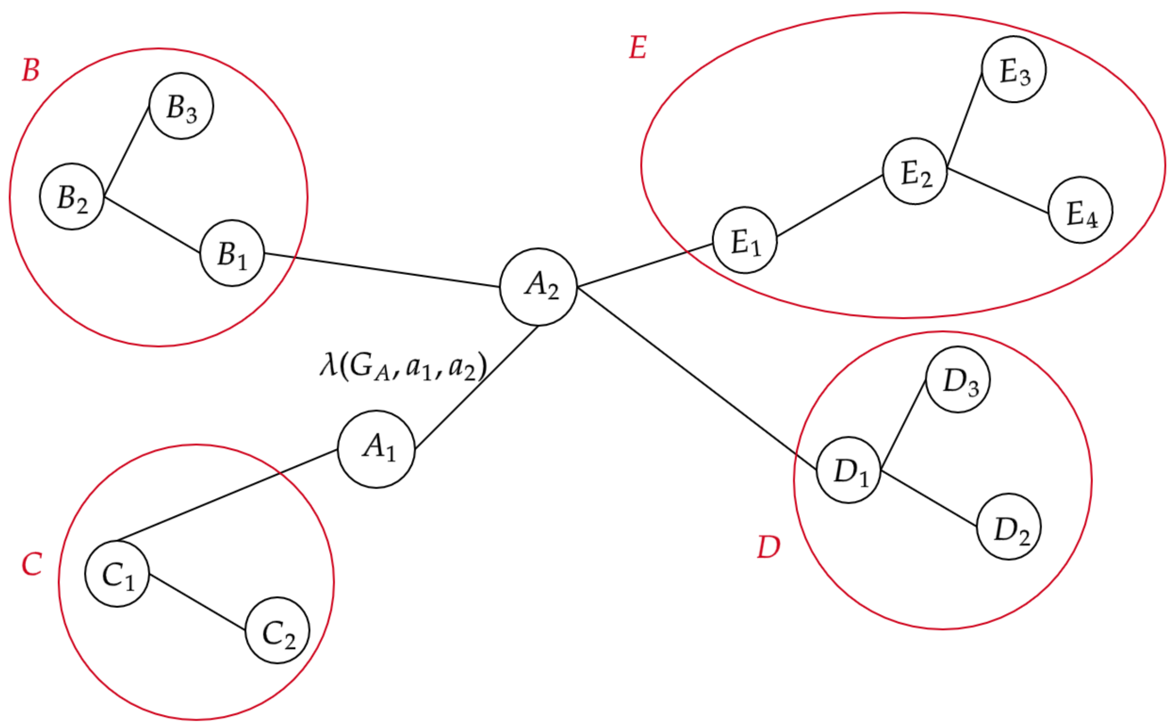


where $|A| \geqslant 2$.

- Pick $a_1, a_2 \in A$

- Contract, in G, the connected component of $T \setminus A$ and let G_A be the resulting graph.
- Compute min a_1-a_2 cut in G_A (say $\delta(X)$, and let $A_1 = X \cap A, A_x = \overline{X} \cap A$).
- Split, in T, A into A_1, A_2, edge A_1-A_2 has label $\lambda(G_A, a_1, a_2)$.

The resulting graph is



Note C_1 gets connected to A_1 if and only if the contracted $v_c \in X$. Same for B_1, E_1, D_1.

We can keep the process above until every vertex in T represents exactly one vertex in G. Let $c, d \in V$, how do I get min c-d cut?

- Get min cost edge e^*in $T_{c,d}$ ($T_{c,d}$ is the only path from c to d in a tree T).
- $\delta(X)$ is a min cost c-d cut where X is one of the component of $T - e^*$.

Note. We need $n - 1$ max flow computations (to construct Gomory-Hu Tree) and a nice data structure to store all v-w cuts.

We should prove the correctness.

Lemma 78

Let $\delta(S)$ be a min r-s cut, and let $v, w \in S$. Then there exists a min v-w cut $\delta(T)$ such that $T \subseteq S$.

Proof. Let $\delta(X)$ be a min v-w cut and note $S \cap X \neq \emptyset, S \cap \overline{X} \neq \emptyset$ by $v \in X, w \in \overline{X}$. By replacing X with $\overline{X}$ and switching r, s if necessary, we may assume $s, w \in S \cap X$.

Case1 $r \in X$. Then since $s \in S \cap X$, we know $r \in X \cap \overline{S}$. Since $u(\delta(A))$ is submodular, we have

$$u(\delta(S)) + u(\delta(\overline{X})) \geqslant u(\delta(S \cap \overline{X})) + u(\delta(S \cup \overline{X}))$$

Note. $u(\delta(A))$ is submodular because,

$$\begin{aligned}
&u(\delta(A \cap B)) + u(\delta(A \cup B)) \\
=&u(E(A \cap B, \overline{A \cup B})) + u(E(A \cap B, A \cup B)) \\
&+ u(\delta(A)) - u(E(A, B \setminus A)) + u(\delta(B)) - u(E(B, A \setminus B)) - u(E(A \cap B, \overline{A \cup B})) \\
=&u(E(A \cap B, B \setminus A)) + u(E(A \cap B, A \setminus B)) \\
&+ u(\delta(A)) - u(E(A, B \setminus A)) + u(\delta(B)) - u(E(B, A \setminus B)) \\
\leqslant&u(\delta(A)) + u(\delta(B))
\end{aligned}$$

Notice that $s \in S \cup \overline{X}$, so it is a r-s cut. so

$$u(\delta(S)) + u(\delta(\overline{X})) \geqslant u(\delta(S \cap \overline{X})) + u(\delta(S \cup \overline{X})) \geqslant u(\delta(S \cap \overline{X})) + u(\delta(S))$$

implies that $\delta(S \cap \overline{X})$ is a min v-w cut with $S \cap \overline{X} \subseteq S$.

Case2 $r \notin X$, analogous.

□

Lemma 79

Let $G = (V, E)$, $u \in \mathbb{R}_+^E$, $s, t \in V$, $B \subseteq V$: $s, t \notin B$. If there exists a min s-t cut $\delta(X)$ with $X \cap B = \emptyset$, then

$$\lambda(G, s, t) = \lambda(G/B, s, t)$$

Proof. Since $X \cap B = \emptyset$, $X \subseteq V(G/B)$. $\lambda(G, s, t) = u(\delta_G(X)) = u(\delta_{G/B}(X)) \geqslant \lambda(G/B, s, t)$. But let $Y \subseteq V(G/B)$ define a min s-t cut in G/B with $v_B \notin Y$ (if $v_B \in Y$, pick $\overline{Y}$). Then we know Y is an s-t cut in G. So

$$\lambda(G/B, s, t) = u(\delta_{G/B}(Y)) = u(\delta_G(Y)) \geqslant \lambda(G, s, t)$$

where $u(\delta_{G/B}(Y)) = u(\delta_G(Y))$ by $v_B \notin Y$. Hence, $\lambda(G, s, t) = \lambda(G/B, s, t)$. □

Definition 80

Suppose T is a Gomory-Hu Tree at any point during the algorithm. Let f_e be its labels for every $e \in E(T)$. Let RS be an edge in T. We say RS has a representative if

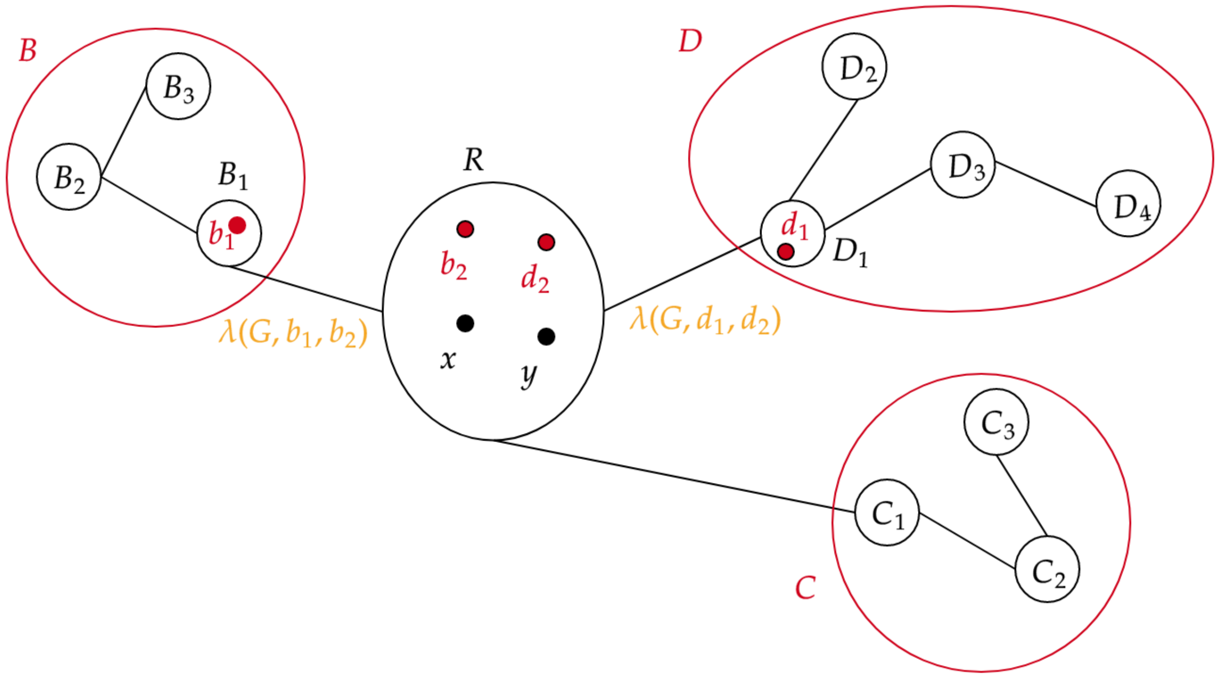
- There exists $r \in R, s \in S : \lambda(G, r, s) = f_{RS}$ **AND**
- The connected component of $T \setminus \{RS\}$ induce the cut of weight $\lambda(G, r, s)$.

Lemma 81

Every edge in $E(T)$ has a representative at all times.

Proof. At first step (when T has only one edge), it is clear, since that is how we create the first edge. Now let $x, y \in R$, and X, Y defining a cut in G_R with $Y = V(G_R) \setminus X$, $x \in X, y \in Y$ and $u(\delta_{G_R}(X)) = \lambda(G_R, x, y)$.

We want to show $\lambda(G_R, x, y) = \lambda(G, x, y)$, so then the two conditions are both satisfied.



Since B_1, R has a representative by our assumption, its label was $\lambda(G, b_1, b_2) = \lambda(G, b_1, b_2)$. Then apply Lemma 79 with $S = \overline{B}$ (where $\overline{B}$ is a min b_1-b_2 cut). There exists min x-y cut $\delta_G(U)$ with $U \subseteq \overline{B}$. Then by Lemma 81, we have $\lambda(G, x, y) = \lambda(G/B, x, y)$.
Since D_1, R has a representative, its label was $\lambda(G, d_1, d_2)$ and $u(\delta_G(D)) = \lambda(G, d_1, d_2)$. Applying Lemma 81, we get $\lambda(G, d_1, d_2) = \lambda(G/B, d_1, d_2)$ because $D \cap B = \emptyset$, so $D \subseteq \overline{B}$. Hence, $\delta_{G/B}(D)$ is still a min d_1-d_2 cut in G/B. Then apply Lemma 79 to get a min x-y cut in G/B, say $\delta_{G/B}(W)$ with $W \subseteq \overline{D}$. Apply Lemma 81 to get

$$\lambda(G, x, y) = \lambda(G/B, x, y) = \lambda((G/B)/D, x, y)$$

Repeat the argument, we get

$$\lambda(G, x, y) = \lambda(G_R, x, y)$$

Let $X' = X \cap R$, $Y' = Y \cap R$, then we know X', Y' has a representative.
Also, we can easily show that the edges where vertex sets did not change (like C_2, C_3) still has a representative, because actually nothing changes for them.
Now, let's say the label of edge B_1, R is $\lambda(G, b, g)$ for a $b \in B$ and $g \in R$. Now, if $g \in X'$, then by definition, the edge B_1, X' still has a representative. But what if $g \in Y'$?

Claim. If $g \in Y'$, $\lambda(G, b, g) = \lambda(G, b, x)$ (show later) which implies B_1, X' has a representative.

Lemma 82

Let $G = (V, E)$, $u \in \mathbb{R}_+^E$, $p, q, r \in V$. Then

$$\lambda(G, p, q) \geqslant \min\{\lambda(G, q, r), \lambda(G, p, r)\}$$

Proof. Let $\delta(M)$ be the min p-q cut. If $r \in M$, then it's also a r-q cut; otherwise, it's a p-r cut, by the definition of λ, we are done. □

Note. Lemma 82 if and only if Smallest two of $\lambda(G, p, q), \lambda(G, q, r), \lambda(G, p, r)$ are equal. For the $\Longleftarrow$ direction, we know the right hand side of Lemma 82 is always the smallest value of the three values, so the inequality holds. For the $\Longrightarrow$ direction, let $\lambda(G, p, q)$ be the smallest, then one of the others must be equal to it, by Lemma 82, done.

Note

$$u(\delta_G(B)) = \lambda(G, b, g) \geqslant \lambda(G, b, x)$$

We've proven $\lambda(G, x, y) = u(\delta_G(S))$ where $S = B \cup X'$. By Lemma 68, there exists a min b-x cut $\delta_G(W)$ with $W \subseteq S$, which means $W \cap Y' = \emptyset$. Now let $G' = G/Y'$, then Lemma 69 implies $\lambda(G, b, x) = \lambda(G', b, x)$. And $g \in Y'$ implies any $v_{Y'}$-b cut in G' isa b-g cut in G implies $\lambda(G', v_{Y'}, b) \geqslant \lambda(G, b, g)$. Also, any x-$v_{Y'}$ cut in G' is an x-y cut in G so $\lambda(G', v_{Y'}, x) \geqslant \lambda(G, x, y)$. Also, the min x-y cut in G is a b-g cut in G by $b \in X'$, $g \in Y'$ which implies $\lambda(G, x, y) \geqslant \lambda(G, b, g)$ Then by Lemma 82, we have

$$\begin{aligned}\lambda(G, b, x) = \lambda(G', b, x) &\geqslant \min\{\lambda(G', v_{Y'}, b), \lambda(G', v_{Y'}, x)\} \\ &\geqslant \min\{\lambda(G, b, g), \lambda(G, x, y)\} = \lambda(G, b, g) \geqslant \lambda(G, b, x)\end{aligned}$$

We prove the claim above, hence, we prove the lemma 81. □

Theorem 83

Let T be final Gomory-Hu tree. Then for every $r, s \in V$, $\lambda(G, r, s)$ is equal to the smallest label of an edge in $T_{r,s}$. Also if e^* is such an edge, then $\lambda(G, r, s) = u(\delta(H))$, where H is one of the connected component of $T \setminus e^*$.

Proof. Let $T_{r,s} = v_0, e_1, v_1, e_2, \ldots, e_k, v_k$, and let f_e be labels of edges in T. By previous result, $\lambda(G, v_{i-1}, v_i) = f_{e_i}, \forall i = 1, \ldots, k$. Show

- $\lambda(G, r, s) \geqslant \min_{i=1,\ldots,k}\{\lambda(G, v_{i-1}, v_i)\}$, which can be easily proved by Lemma 82

- Then we pick the e^* as described above and r is in one component of $T \setminus e^*$ and s is in the other. Thus we find a r-s cut H where

$$\lambda(G, r, s) \leqslant u(\delta(H)) = \min_{i=1,\ldots,k} \{\lambda(G, v_{i-1}, v_i)\} = f_{e^*}$$

  Hence, $\lambda(G, r, s) = u(\delta(H)) = f_{e^*}$ as required.

$\square$