## Part 9 Steiner Forest Problem

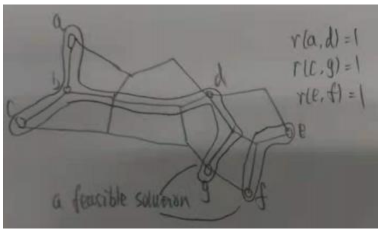
(**Definition 1**) (**Steiner Forest Problem**, SFP) Given a graph $G$=($V$, $E$), edge cost $c$: $E \to R^+$ (associated with each edge), and several vertex sets $S_i \subseteq V$ (where all the vertices in a set $S_i$ are **connected**) , find a **minimum cost subgraph** $F$ such that each pair of vertices belonging to the same set $S_i$ are connected. Note that vertices in two different sets $S_i$ and $S_j$ can be connected as well, but any two sets must be disjoint, i.e., $S_i \cap S_j = \varnothing$.

**Problem Restatement**: Define a **connectivity requirement function** $r$ that maps unordered pairs of vertices to {0, 1} as follow:

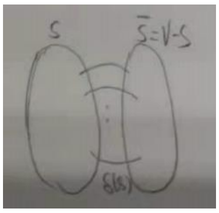
$$r(u,v) = \begin{cases} 1, & u,v \in S_i \\ 0, & \text{otherwise} \end{cases}.$$

Note that $r(u, v)$=0 doesn't mean $u$ and $v$ aren't connected (i.e., there is no path between $u$ and $v$) in the original graph $G$. In fact, $r(u, v)$=1 means that we want that there is a path between $u$ and $v$.

Notes:




(**Definition 2**) (**ILP & LP-Relaxation** for **SFP**) For any set $S \subseteq V$, let $\bar{S}$ denote ($V$-$S$). Let $\delta(S)$ denote the set of edges with exactly one end point in $S$. Alternatively, $\delta(S)$ represents the set of edges that 'cross' the cut $(S,\bar{S})$.




We associate each **edge** $e \in E$ with a **binary variable** such that

$$x_e = \begin{cases} 1, & e \in F \\ 0, & \text{otherwise} \end{cases}.$$

**SFP** can be reformulated as the following **ILP**:

$$\begin{aligned} &\min \sum_{e \in E} c_e x_e \\ &\text{s.t.} \sum_{e \in \delta(S)} x_e \ge 1, \forall S \in S^*, \\ &x_e \in \{0,1\}, \forall e \in E \end{aligned}$$

where $S^*$ is the **collection of all sets** $S$ such that $(S,\bar{S})$ separates each vertex pair ($u$, $v$) with $r(u, v)$=1. Note that if $S \in S^*$, then $\bar{S} \in S^*$.

Consider any cut $(S,\bar{S})$ in $G$ that separates a vertex pair ($u$, $v$), i.e., (1) $u \in S$ and $v \in \bar{S}$ or (2) $v \in S$ and $u \notin \bar{S}$. If $r(u, v)$=1, then we must pick at least one edge $e \in \delta(S)$. Clearly, this is **necessary** and also **sufficient** (proved in **Lemma 1**).

For the aforementioned ILP, we have the corresponding **LP-Relaxation**:

$$\begin{aligned} &\min \sum_{e\in E} c_e x_e \\ &\text{s.t.} \sum_{e\in\delta(S)} x_e \geq 1, \forall S \in S^* . \\ &x_e \geq 0, \forall e \in E \end{aligned}$$

Note: For both the **ILP** and **LP-Relaxation**, there're <u>$|E|$ variables</u>, but <u>the number of constraints is exponential</u>. However, we don't need to really solve the LP when we consider a **Primal-Dual Algorithm** for **SFP**.

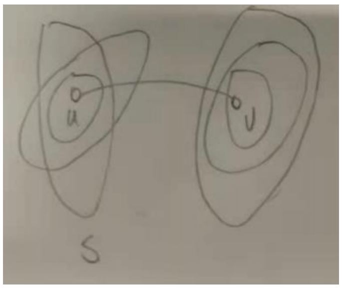
(**Lemma 1**) A vertex pair ($u$, $v$) is **connected** if and only if for all cuts $(S,\bar{S})$ that separate ($u$, $v$), there's an edge $e \in \delta(S)$.

**Proof** of **Lemma 1**. (**Converse-Negative Proposition**) Suppose that $u$ and $v$ are **not connected**. Let $S$ be the set of vertices reachable from $u$ (which can be derived by applying BFS/DFS to $u$). Then, $v$ must be in $\bar{S}$ and there is no edge between $(S,\bar{S})$. It indicates that for a vertex set $S$ if there is no edge between $(S,\bar{S})$, then any vertex pair ($u$, $v$) separated by $(S,\bar{S})$ is not connected, which complete the proof.

(**Definition 3**) (**Dual LP** for **SFP**) Treat the **LP-Relaxation** of **SFP** (in **Definition 2**) as the **Primal LP**, we have the following **Dual LP**:

$$\begin{aligned} &\max \sum_{S\in S^*} y_S \\ &\text{s.t.} \sum_{S\in S^*: e\in\delta(S)} y_S \leq c_e, \forall e \in E , \\ &y_S \geq 0, \forall S \in S^* \end{aligned}$$

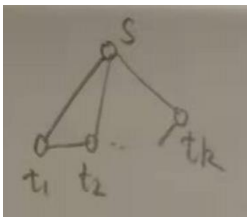
where we associate each set $S \in S^*$ with a variable $y_S$. Consider the constraint $\sum_{e\in\delta(S)} y_S \leq c_e$ associated with an edge $e$=($u$, $v$), there should be multiple set $S \in S^*$ with each one associated with a weight ('charge') $y_S$. And the sum of all the weight $y_S$ is at most $c_e$.




Note: The number of dual variables is exponential, but this will not bother us, because <u>only a polynomial number of dual variables will be non-zero</u> (i.e., most of the dual variables are zero).

(**Example 1**) Consider a <u>**complete graph**</u> with ($k$+1) vertices. All edges have <u>unit weight</u>. Let the **connectivity requirement function** be $r(u, v)$=1 for all $u,v \in V$. Let $OPT_{\text{D-LP}}(I)$, $OPT_{\text{LP}}(I)$ and $OPT_{\text{ILP}}(I)$ be the optimal value of the **Dual LP**, **LP Relaxation**, and **Primal ILP** for **SFP**. By **Strong Duality**, we have $OPT_{\text{D-LP}}(I)=OPT_{\text{LP}}(I)$.

Without the loss of generality, for an arbitrary vertex $s \in V$, we denote the rest $k$ vertices as $\{t_1, t_2, \cdots, t_k\}$.




For the **ILP**, to satisfy all the connectivity requirements (i.e., $r(s,t_i)=1$ for $i \in \{1,2,\cdots,k\}$), $k$ edges should be selected. Hence, we have $OPT_{\text{ILP}}(I)=k$.

For the **Dual LP**, consider the set $\{s\}$ and we have $\{s\} \in S^*$. If we raise the 'charge' of $\{s\}$ to 1, i.e., $y_{\{s\}}=1$, then all the edges $\{(s,t_1),(s,t_2),\cdots,(s,t_k)\}$ become '**tight**'. In this case, we cannot raise the 'charge' of other sets contained $\{t_1,t_2,\cdots,t_k\}$, e.g., $y_{\{t_1\}}$, $y_{\{t_2\}}$, etc. (where $\{t_1\} \in S^*$, $\{t_2\} \in S^*$, etc.). Thus, we have found a **feasible solution** to the **Dual LP** and we know $OPT_{\text{D-LP}}(I) \geq 1$.

In summary, the **approximation ratio** is bounded by

$$\frac{OPT_{\text{ILP}}(I)}{OPT_{\text{LP}}(I)} = \frac{OPT_{\text{ILP}}(I)}{OPT_{\text{D-LP}}(I)} \leq k,$$

which is not a good bound for the approximation ratio. In fact, we can further improve the lower bound of $OPT_{\text{D-LP}}(I)$ (i.e., get larger lower bound of $OPT_{\text{D-LP}}(I)$) by finding a better **feasible solution** to the **Dual LP**.

Furthermore, consider another strategy that raise the 'charge' of the sets $\{\{s\},\{t_1\},\{t_2\},\cdots,\{t_k\}\}$ simultaneously. For an arbitrary edge $(s, t_i)$, it becomes '**tight**' when we simultaneously raise the 'charge' of $\{s\}$ and $\{t_1\}$ to 0.5, i.e., $y_{\{s\}} = y_{\{t_i\}} = 0.5$. In this case, all the other edges (e.g., $(t_i, t_j)$) become 'tight'. Thus, we have $OPT_{\text{D-LP}}(I) \geq (k+1)/2$. The corresponding **approximation ratio** is bounded by

$$\frac{OPT_{\text{ILP}}(I)}{OPT_{\text{LP}}(I)} = \frac{OPT_{\text{ILP}}(I)}{OPT_{\text{D-LP}}(I)} \leq 2(1+\frac{1}{k}) \approx 2 \cdot$$

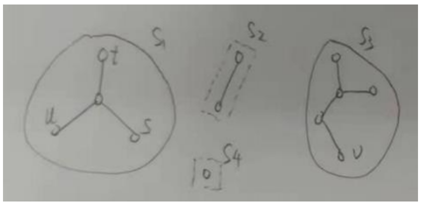
Notes: The latter example gives a new idea to design the **Primal-Dual Algorithm** for **SFP**. Namely, we can raise **duals** $\{y_S\}$ in a synchronized manner instead of trying to satisfy a single unsatisfied primal constraint. We can try out many possibilities at the same time.

(**Algorithm 1**) (**Primal-Dual Algorithm**) We say that an **edge** $e$ '**feels**' a **dual variable** $y_S$ if $y_S>0$ and $e \in \delta(S)$. We say that an **edge** $e$ is '**tight**' if the total amount of **duals** it feels equals its cost, i.e., $\sum_{e \in \delta(S): S \in S^*} y_S = c_e$. We say that a **set** $S$ has been '**raised**' if $y_S>0$.

The **Dual LP** tries to maximize the sum of the dual variables $\{y_S\}$ subject to the condition that no edge is '**over-tight**', i.e., no edge feels more dual than it cost. **Degree** of **set** $S$ is defined as the number of picked edges crossing the cut $(S, \bar{S})$.

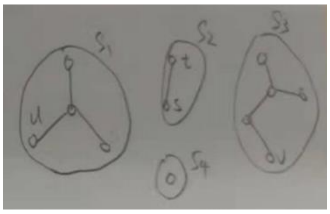
We use $F$ to indicate the set of edge picked. We say that a **set** $S$ is **unsatisfied** if $S \in S^*$, but there're no picked edges crossing the cut $(S, \bar{S})$. Clearly, if $F$ is not **Primal Feasible**, there must be a **connected component** in $F$ that is **unsatisfied**. We say such a **connected component** is '**active**'.

For example, consider the following **forest** $F$ with 4 **connected components**. Suppose that we only have 2 **connectivity requirements**: $r(u, v)=1$ and $r(s, t)=1$.




$S_1$ and $S_3$ are **unsatisfied**, because we have $r(s, t)=1$ (i.e., $S_1, S_2 \in S^*$) but there're no edges crossing $S_1$ and $S_3$. In contrast, $S_2$ and $S_4$ are already **satisfied**, where we know $S_2 \notin S^*$ and $S_4 \notin S^*$ according to the 2 connectivity requirements.

Note that if we revised the aforementioned example as follow, we still have $S_2 \notin S^*$ (w.r.t. the connectivity requirement $r(s, t)=1$):

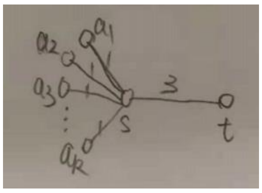



**Initialization**: Initially, we set $x_e=0$ for each edge $e \in E$, so we have $F = \varnothing$. For each set $S \in S^*$, we set $y_S=0$.

**Iterative Procedure**: In each iteration, we raise the **dual variables** for all the **active component** in a synchronized manner, until some edges go '**tight**'. We then pick one of these '**tight**' **edges** and go to the next iteration.

We stop the iterative procedure when a **primal feasible solution** $F$ is found.

(**Example 1**) Consider the following graph with only 1 connectivity requirement $r(s, t)=1$ for **Algorithm 1**:




Initially, each single vertex forms a connected component. There're 2 **active components** $\{s\}$ and $\{t\}$. In the 1st iteration, we simultaneously raise the **duals** of $\{s\}$ and $\{t\}$ by 1 unit. Especially, the edge $(s, t)$ feels 2 sets $\{s\}$ and $\{t\}$, but it isn't 'tight'. All the rest edges $\{(s, a_i)\}$ feel only 1 set $\{s\}$ and are all 'tight'.

According to **Algorithm 1**, we need to pick one of the 'tight' edges. Suppose we pick the edge $(s, a_1)$. Then, we have 2 **active components** $\{s, a_1\}$ and $\{t\}$. We cannot raise the **duals** for $\{s, a_1\}$ and $\{t\}$, because edges $\{(s,a_2),(s,a_3),\cdots,(s,a_k)\}$ are already '**tight**'.

We need to pick one of the 'tight' edges. Suppose we pick the edge $(s, a_2)$. Then, we still have 2 **active components** $\{s, a_1, a_2\}$ and $\{t\}$. And we cannot raise the **duals** for them.

Finally, we pick all the edges $\{(s,a_1),(s,a_2),\cdots,(s,a_k)\}$ and still have 2 **active components** $\{s,a_1,a_2,\cdots,a_k\}$ and $\{t\}$.

In the last iteration, we raise the **duals** for $\{s,a_1,a_2,\cdots,a_k\}$ and $\{t\}$ by 0.5 unit. Then, we pick the **'tight' edge** ($s$, $t$). In this case, we obtain a feasible solution $F=\{(s,t),(s,a_1),(s,a_2),\cdots,(s,a_k)\}$. Let $c(F)$ be the cost of such a feasible solution. We have $c(F)=3+k$, which is the **upper bound** of $c(F)$.

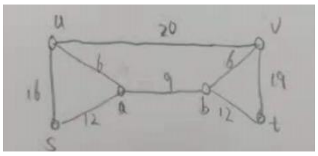
For the aforementioned example, the optimal cost is $OPT_{\mathrm{ILP}}(I)=3$, which equals the total sum of duals we raised (i.e., 2×1+2×0.5=3). It's the **lower bound** of $c(F)$ (if we pick the 'tight' edge ($s$, $t$) in the 1st iteration).

In summary, **Algorithm 1** has a good lower bound but bad upper bound.

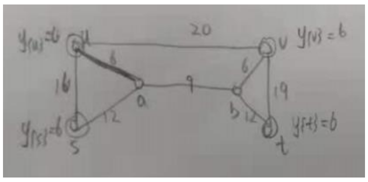
(**Algorithm 2**) (**Modified Primal-Dual Algorithm**) For the **feasible solution** $F$ given by **Algorithm 1**, it may contain **redundant edges** (e.g., $\{(s,a_1),(s,a_2),\cdots,(s,a_k)\}$ in **Example 1**). We say an edge $e\in F$ is **redundant** if $F$-$\{e\}$ is still feasible.

All the **redundant edges** can be dropped simultaneously from $F$. Let $F'$ denote the final forest after pruning, which is also the output of the **modified Primal-Dual Algorithm**.

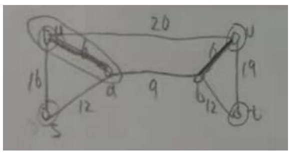
(**Example 2**) Consider the following example graph with only 2 connectivity requirements $r(s, t)=1$ and $r(u, v)=1$:



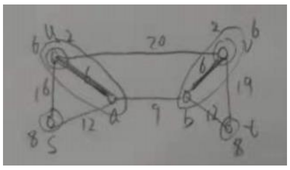

In iteration #1, we have 4 active components $\{s\}$, $\{t\}$, $\{u\}$, and $\{v\}$. We raise the duals of $\{s\}$, $\{t\}$, $\{u\}$, and $\{v\}$ by 6. Thus, we have $y_{\{s\}}=y_{\{t\}}=y_{\{u\}}=y_{\{v\}}=6$. Edges ($u$, $a$) and ($v$, $b$) become tight. Suppose we pick the tight edge ($u$, $a$).



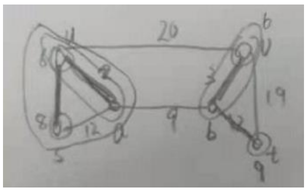

In iteration #2, we have 4 active components $\{s\}$, {t}, $\{u, a\}$, and $\{v\}$. Since edge ($v$, $b$) is tight, we don't raise the duals but pick the tight edge ($u$, $a$).



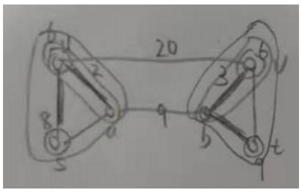

In iteration #3, we have 4 active components $\{s\}$, $\{t\}$, $\{u, a\}$, and $\{v, b\}$. We raise the duals of $\{s\}$, $\{t\}$, $\{u, a\}$, and $\{v, b\}$ by 2. Then, we have $y_{\{s\}}=y_{\{t\}}=8$, $y_{\{u\}}=y_{\{v\}}=6$, $y_{\{u,a\}}=y_{\{v,b\}}=2$. Edge ($u$, $s$) becomes tight. We pick the tight edge ($u$, $s$).



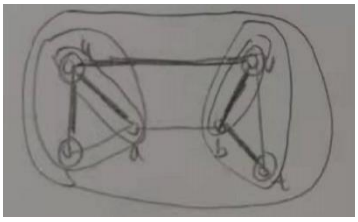

In iteration #4, we have 3 active components $\{s, u, a\}$, $\{v, b\}$, and $\{t\}$. We raise the duals of $\{s, u, a\}$, $\{v, b\}$, and $\{t\}$ by 1. Then, we have $y_{\{s\}}= 8$, $y_{\{u\}}=y_{\{v\}}=6$, $y_{\{t\}}=9$, $y_{\{u, a\}}=2$, $y_{\{v, b\}}=3$. Edge $(b, t)$ becomes tight. We pick the tight edge $(b, t)$.




In iteration #5, we have 2 active components $\{s, u, a\}$ and $\{t, v, b\}$. We raise the duals of $\{s, u, a\}$ and $\{t, v, b\}$ by 1.5. Then, we have $y_{\{s\}}= 8$, $y_{\{u\}}=y_{\{v\}}=6$, $y_{\{t\}}=9$, $y_{\{u, a\}}=2$, $y_{\{v, b\}}=3$, $y\{s, u, a\}=y\{t, v, b\}=1.5$. Edge $(u, v)$ becomes tight. We pick the tight edge $(u, v)$.




In iteration #6, there're no active components. We have a feasible forest $F=\{(s, u), (t, b), (u, v), (u, a), (v, b)\}$.



We can drop (prune) one edge $(u, a)$ and derive a feasible forest $F'=\{(s, u), (t, b), (u, v), (v, b)\}$.

The **cost** of such a **feasible solution** is $c(F')$=16+20+6+12=54.

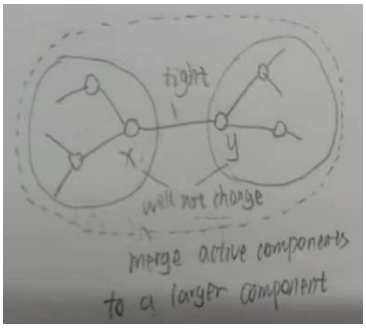
In contrast, the objective value of the **Dual LP** is $y_{\{s\}}+y_{\{u\}}+y_{\{v\}}+y_{\{t\}}+y_{\{u, a\}}+y_{\{v, b\}}+y_{\{s, u, a\}}+y_{\{t, v, b\}}$=8+6×2+9+2+3+1.5×2=37, which gives a **lower bound** for the cost of Stiner Forest.

The **optimal solution** to the aforementioned SFP is $\{(u, a), (s, a), (a, b), (v, b), (t, b)\}$. Thus, the **optimal cost** of the Stiner Forest is 6+12+9+6+12=45.

Note that the approximation factor of **Algorithm 2** in **Example 2** is better than (≤) 54/37. In fact, the **approximation factor** of **Algorithm 2** is always less than or equal to 2.

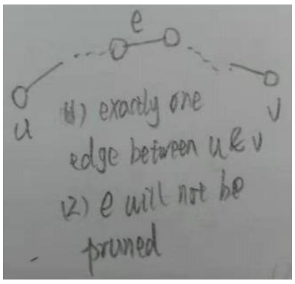
(**Claim 1**) The **Dual solution** (i.e., $\{y_S \mid \forall S \in S^*\}$) and **Primal solution** (i.e., the picked edges) derived by **Algorithm 2** are both **feasible** at termination.

**Proof** of **Claim 1**. In each iteration of **Algorithm 2**, we raise the dual variables w.r.t. active components util some edges become **tight**. We then select a tight edge and all the dual variables felt by the tight edge are frozen (i.e., will not change in the rest iterations). Hence, the **Dual solution** derived by **Algorithm 2** is feasible, i.e., $\{y_S \mid \forall S \in S^*\}$ satisfy all the tight constraints in Dual LP.




Before the **pruning step** of **Algorithm 2**, the **Primal solution** is clearly feasible, because we stop the algorithm only when all the components of the forest are inactive, i.e., all the **connectivity requirements** are satisfied.

Consider any two vertices $u$ and $v$ with **connectivity requirement** $r(u, v)=1$. There's exactly one unique path between $u$ and $v$ in the derived forest before the **pruning step**, because in each iteration we merge smaller components (which are forests) into a larger component, which ensures that the derived result is always a forest (without any circles).




In the **pruning step**, we try to drop some edges while ensuring the connectivity requirements in the forest, so all the edges in the path between $u$ and $v$ will not be deleted. Thus, the **pruning step** does no 'damage' to the connectivity of the **Primal solution**.

In summary, the **Primal solution** derived by **Algorithm 2** is also feasible.

(**Theorem 1**) Let $OPT_{\text{SFP}}(I)$ be the cost of optimal Steiner forest. Let $F'$ be the final Stiner forest given by **Algorithm 2** (i.e., after the pruning step) and $cost(F')$ be the cost of $F'$. The approximation ratio of **Algorithm 2** satisfies

$$\frac{cost(F')}{OPT_{\text{SFP}}(I)} \le 2 \cdot$$

**Proof** of **Theorem 1**. By the **Weak Duality Theorem**, we have

$$\sum_{S \in S^*} y_S \le OPT_{\text{SFP}}(I) \cdot$$

Let $\Delta_i$ be the **amount** by which the **dual variable** of each **active component** is raised in the $i$-th iteration. Let $c_i$ be the number of **active components** in the $i$-th iteration. For the dual variables, we also have

$$\sum_{S \in S^*} y_S = \sum_i \Delta_i \times c_i \cdot$$

Let $d_i$ be the sum of **degree** of all **active components** in the $i$-th iteration and $\bar{d}_i$ be the **average degree** of all **active components** in the $i$-th iteration. For the cost of $F'$, we have

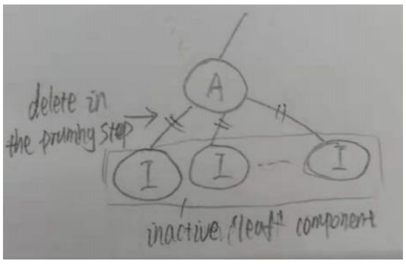
$$cost(F') = \sum_i (\Delta_i \times d_i) = \sum_i (\Delta_i \times \bar{d}_i \times c_i) \cdot$$

Note that a **tree** with $n$ vertices has ($n$-1) edges. Thus, the **average degree** of a tree is

$$\frac{2(n-1)}{n} = 2 - \frac{2}{n} < 2 \cdot$$

Further, for a **forest** with $n$ vertices has at most (≤) ($n$-1) edges. Thus, the average degree of a forest is also less than 2. Such a condition also holds for connected components in a forest.

In the $i$-th iteration of **Algorithm 2**, there may be both **active components** and **inactive components**. Consider the special case when **inactive components** could be '**leaves**':




In this case, the **average degree** of all **active components** will not be less than 2. Note that the edges connected to these **inactive 'leaves'** will be deleted in the pruning step. Thus, we can ensure that $\bar{d}_i < 2$.

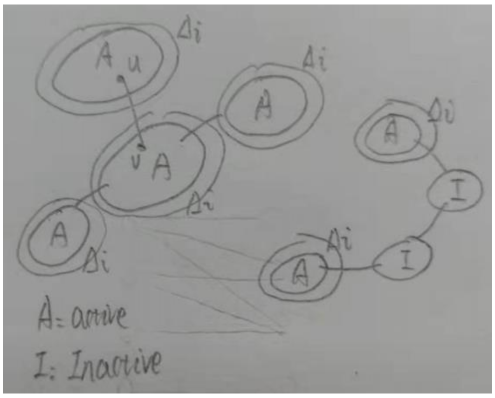
For the cost of $F$', we further have

$$cost(F') = \sum_i (\Delta_i \times \bar{d}_i \times c_i) < \sum_i (2\Delta_i \times c_i) = 2\sum_{S \in S^*} y_S \cdot$$

Since we already have $\sum_{S \in S^*} y_S \le OPT_{\text{SFP}}(I)$, we can derive an upper bound of the **approximation ratio** of **Algorithm 2**:

$$\frac{cost(F')}{OPT_{\text{SFP}}(I)} < \frac{2\sum_{S \in S^*} y_S}{OPT_{\text{SFP}}(I)} =\le \frac{2 \cdot OPT_{\text{SFP}}(I)}{OPT_{\text{SFP}}(I)} = 2 \cdot$$

Notes: Suppose in the $i$-th iteration of **Algorithm 2**, we have the following connected components (including **active** and **inactive** components):




The total degree of active components is 1+1+1+3+1+1=8. We simultaneously raise the duals of all the active components (by $\Delta_i$) until some edges become tight, e.g., ($u$, $v$).