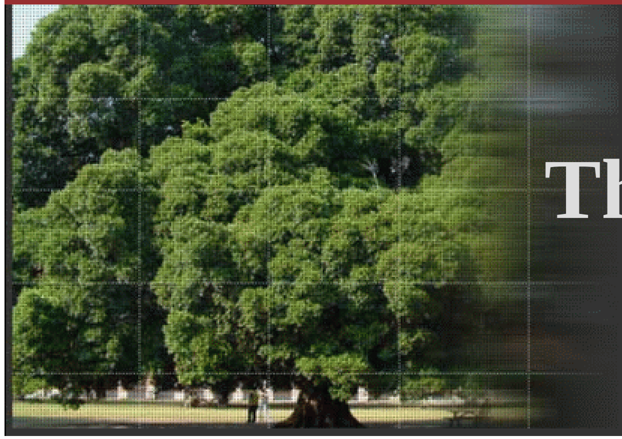
National Cheng Kung University



The Internal Steiner Tree Problem

Sun-Yuan Hsieh (謝孫源)

Department of Computer Science and Information Engineering

Institute of Medical Informatics

Institute of Manufacturing Information and Systems

National Cheng Kung University

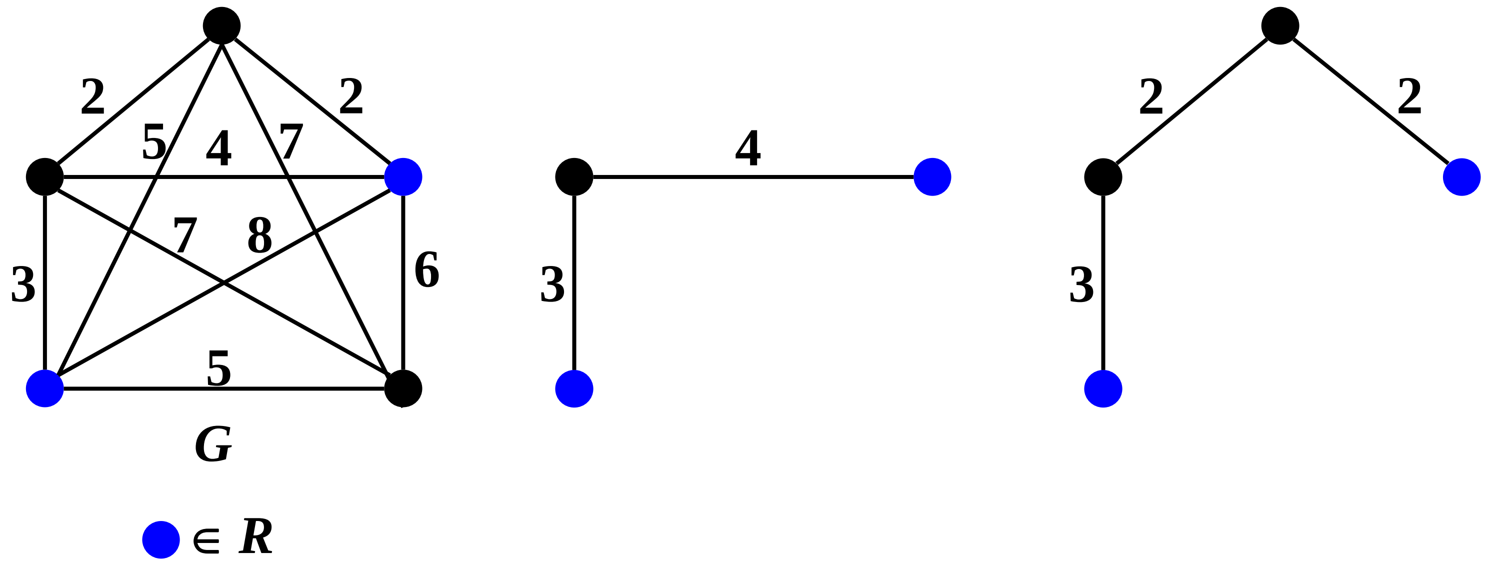
hsiehsy@mail.ncku.edu.tw

http://algorithm.csie.ncku.edu.tw

Outline

- **Introduction**
  - **What is the Steiner tree?**
  - **applications of the Steiner tree**
  - **variants of the Steiner tree**
- **MAX SNP-hardness**
- **Approximation algorithm**
- **ISTP(1,2)**
- **Concluding remarks**

- What is the Steiner tree?
  - Given a graph $G = (V, E)$ with a cost (or distance) function c: $E \rightarrow \mathbf{R}^+$, and a vertex subset $R \subseteq V$, a ***Steiner tree*** is a connected and acyclic subgraph contains all the vertices in R (**terminals**).

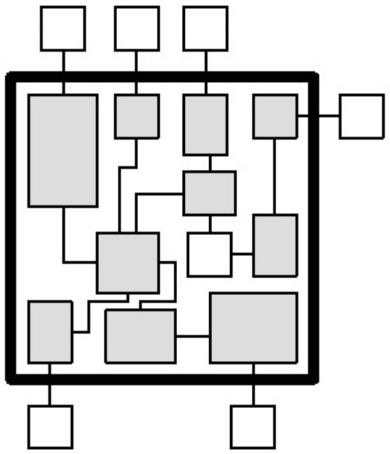


Introduction (2/11)

- **Steiner tree problem**
  - **the decision version: NP-complete [Karp, 1972 Complexity of Computer Computations]**
  - **in the Euclidean metric: NP-complete [Garey *et al*., 1977 SIAM J. APPL. MATH.]**
  - **in the rectilinear metric: NP-complete [Garey and Johnson, 1977 SIAM J. APPL. MATH.]**
  - **MAX SNP-hard [Bern and Plassmann, 1989 IPL]**
  - **1.55-approximation algorithm [Robins and Zelikovsky, 2000 SODA]**
  - **1.39-approximation algorithm [Byrka *et al*., 2010 STOC]**
  - **1.25-approximation algorithm for STP(1, 2) [Berman *et al*., 2009 WADS]**

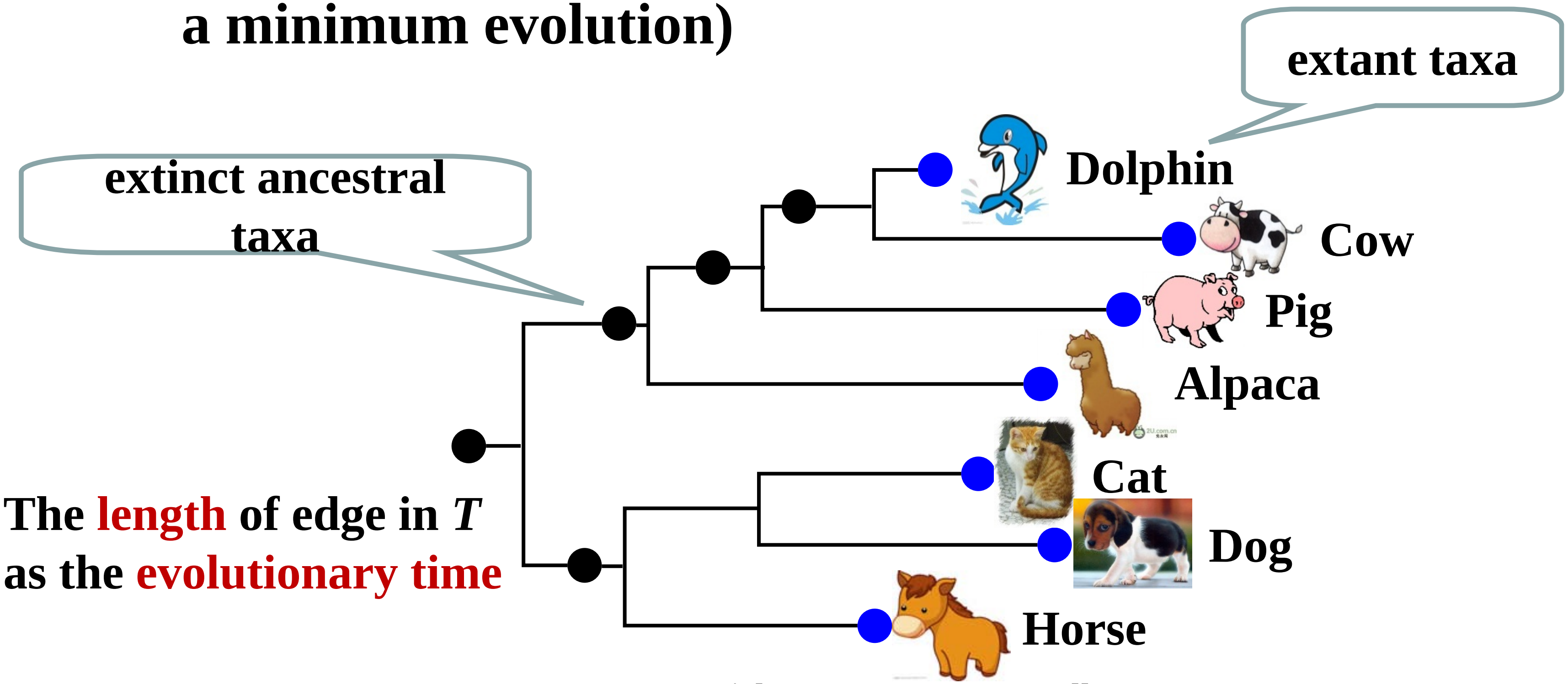
Introduction (3/11)

- Applications
  - VLSI design
    - local or global routing
    - placement problem
    - engineering change order (ECO)



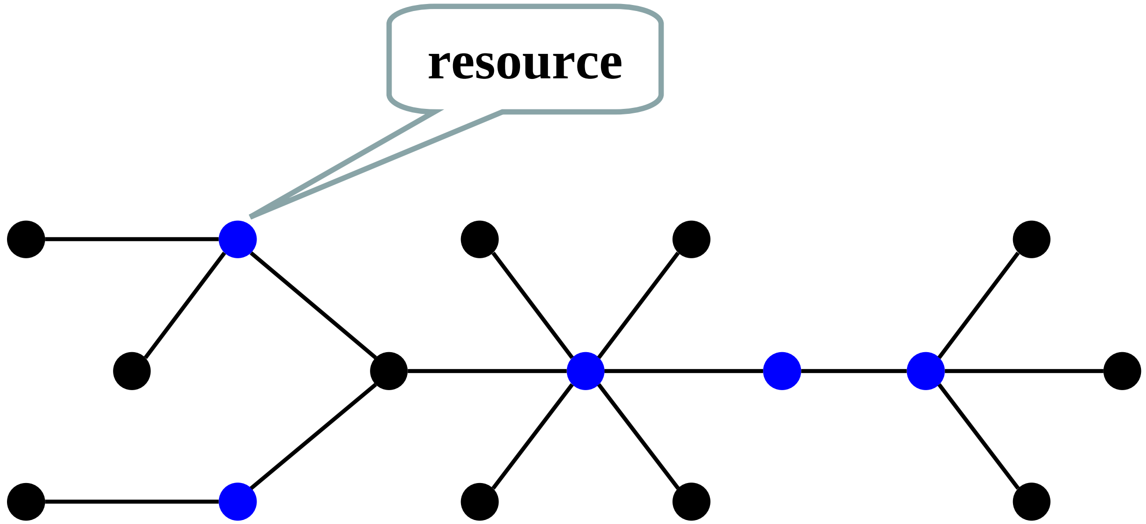
Introduction (4/11)

- Applications
  - computational biology - reconstruction of phylogenetic
    - **minimize the tree length** according to the principle of parsimony (i.e., nature always finds the paths that require a minimum evolution)



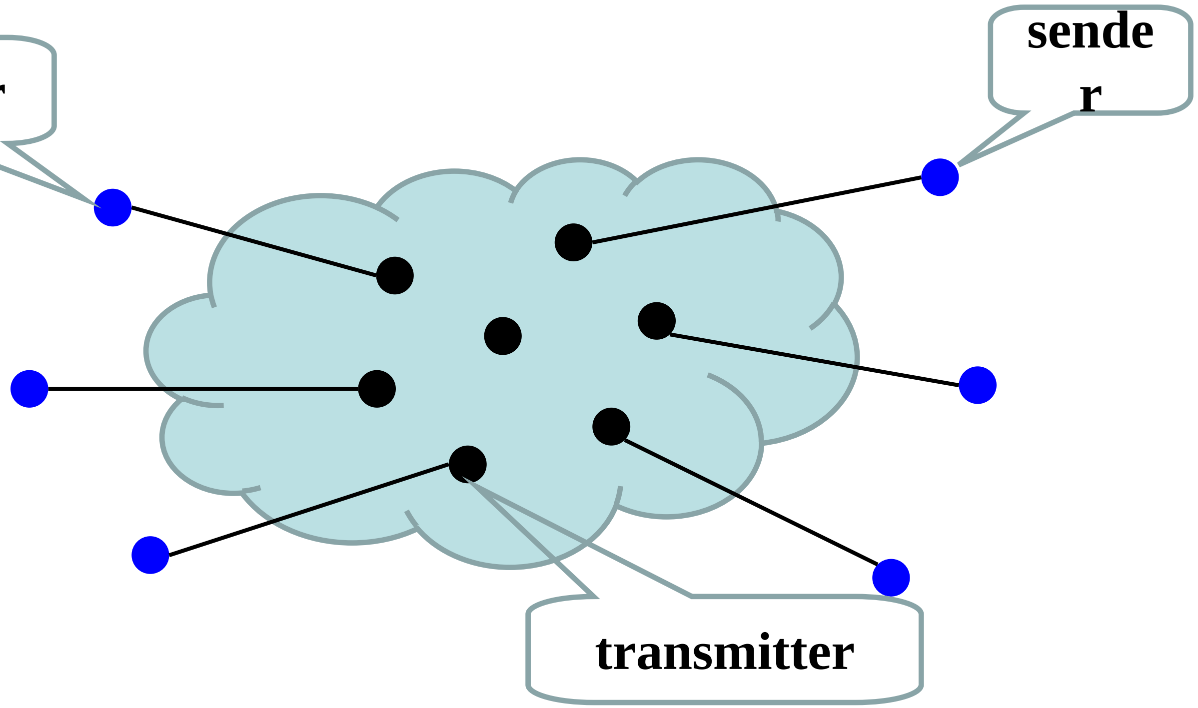
The **length** of edge in *T* as the **evolutionary time**

- Applications
  - network routing - resource allocation



Introduction (6/11)

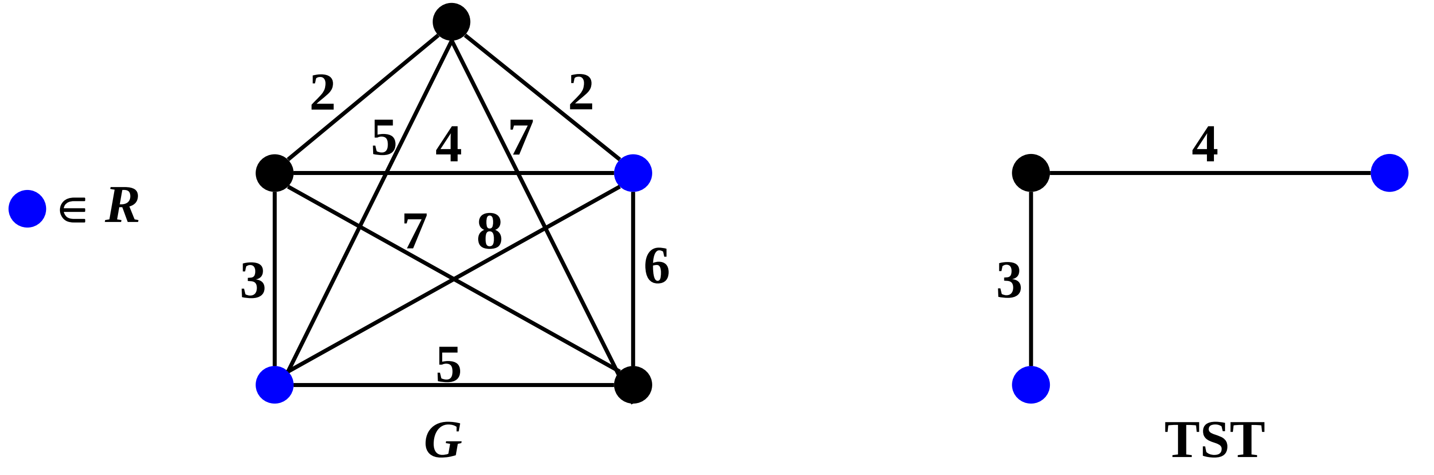
- Applications
  - telecommunications



  - wireless communications, transportation

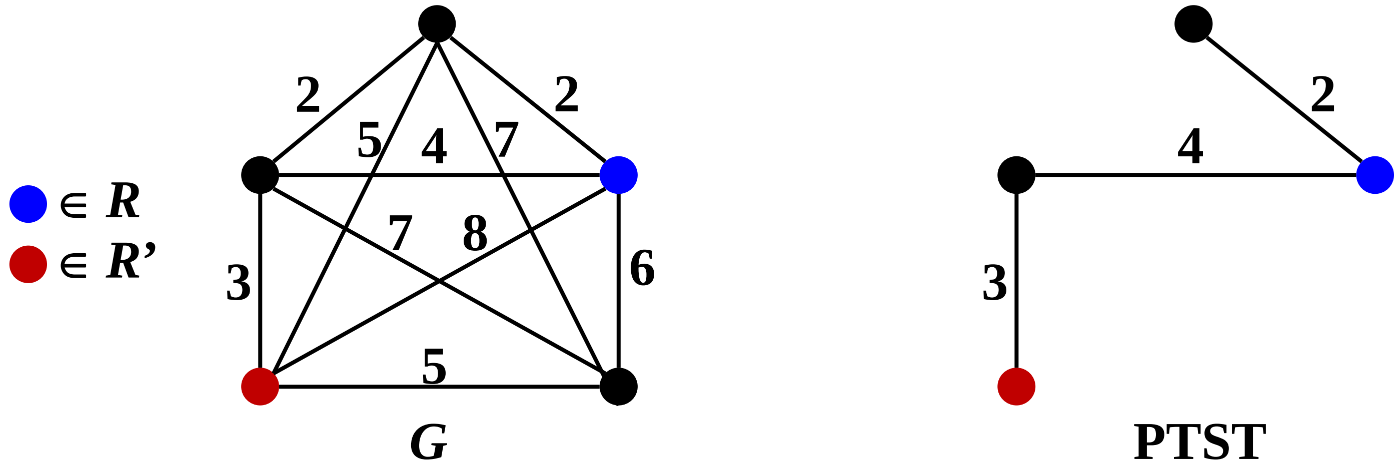
Introduction (7/11)

- terminal Steiner tree problem (TSTP): also called full Steiner tree problem
  - each vertex in R must be a leaf
  - NP-complete and MAX SNP-hard [Lu *et al.*, 2003 TCS]
  - $(2\rho - \frac{\rho}{3\rho-2})$-approximation algorithm [Martinez *et al.*, 2007, TCS]
  - 1.42-approximation algorithm for TSTP(1, 2) [Martinez *et al.*, 2007, TCS]



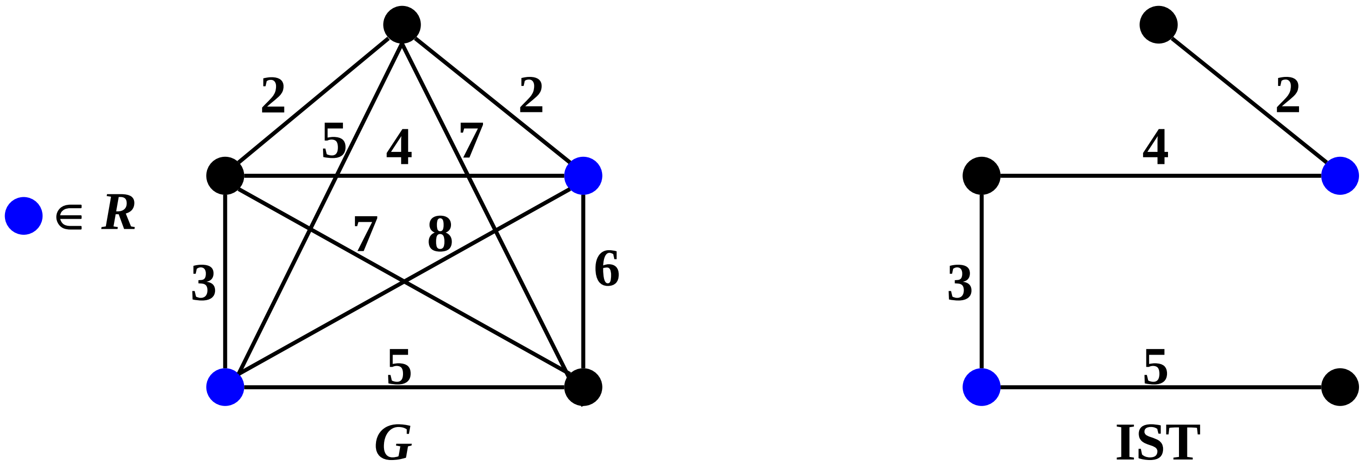
Introduction (8/11)

- **partial terminal** Steiner tree problem (**PTSTP**): $R' \subseteq R \subset V$
  - each vertex in R' must be a **leaf**
  - NP-complete and MAX SNP-hard **[Hsieh and Gao, 2007 JS]**
  - $(2\rho - \frac{\rho}{3\rho-2} - \varepsilon)$-approximation algorithm **[Hsieh and Pi, 2008 ISPAN]**



Introduction (9/11)

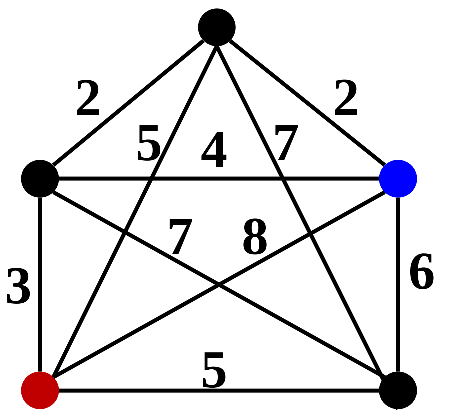
- internal Steiner tree problem (ISTP):
  - each vertex in R must be an internal vertex
  - NP-complete and MAX SNP-hard [Huang *et al.*, 2013 JOCM]
  - $(2\rho + 1)$-approximation [Huang *et al.*, 2013 JOCM]
  - $\frac{9}{7}$-approximation algorithm for ISTP(1,2) [Huang *et al.*, 2013 JOCM]



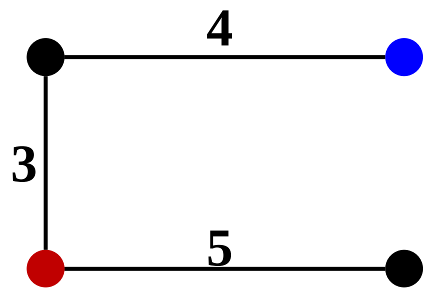
Introduction (10/11)

- **Selected Internal Steiner tree problem (SISTP): $R' \subset R \subseteq V$**
  - **each vertex in R' must be an internal vertex**
  - **NP-complete and MAX SNP-hard [Hsieh and Yang, 2007 TCS]**
  - **$(1 + \rho)$-approximation algorithm [Li *et al.*, 2010 Algorithmica]**

$\in R$
$\in R'$



G



SIST

Introduction (11/11)

	NP-completeness	MAX SNP-hardness	Best approximation ratio
STP	[Karp, 1972]	[Bern, 1989]	$\ln 4 + \varepsilon < 1.39$ [Byrka *et al.*, 2010]
STP(1,2)			1.25 [Berman *et al.*, 2009]
TSTP	[Lu *et al.*, 2003]	[Lu *et al.*, 2003]	$2\rho - \frac{\rho}{3\rho-2}$ [Martinez *et al.*, 2007]
TSTP(1,2)			1.42 [Martinez *et al.*, 2007]
PTSTP	[Hsieh *et al.*, 2007]	[Hsieh *et al.*, 2007]	$2\rho - \frac{\rho}{3\rho-2} - \varepsilon$ [Hsieh *et al.*, 2008]
ISTP	[Huang *et al.*, 2013]	[Huang *et al.*, 2013]	$2\rho + 1$ [Huang *et al.*, 2013]
ISTP(1,2)			$\frac{9}{7}$ [Huang *et al.*, 2013]
SISTP	[Hsieh *et al.*, 2007]	[Hsieh *et al.*, 2007]	$1+\rho$ [Li *et al.*, 2010]

MAX SNP-hardness (1/11)

- Any MAX SNP-hard problem has a PTAS, then P = NP.
  - Journal of the Association for Computing Machinery 1998.
- How to show a problem to be MAX SNP-hard: by providing an *L*-reduction from some MAX SNP-hard problem to it.

- Lemma 1. STP(1, 2) is MAX SNP-hard.
  - Bern and Plassmann, Information Processing Letters 32(4), 1989.
- We prove that ISTP is MAX SNP-hard by providing an *L*-reduction from STP(1, 2) to ISTP.

MAX SNP-hardness (2/11)

- **L-reduction:**

  P_1 and P_2 : two optimization problems with input instances I_1 and I_2

  P_1 *L*-reduces to P_2 if the following conditions are satisfied:

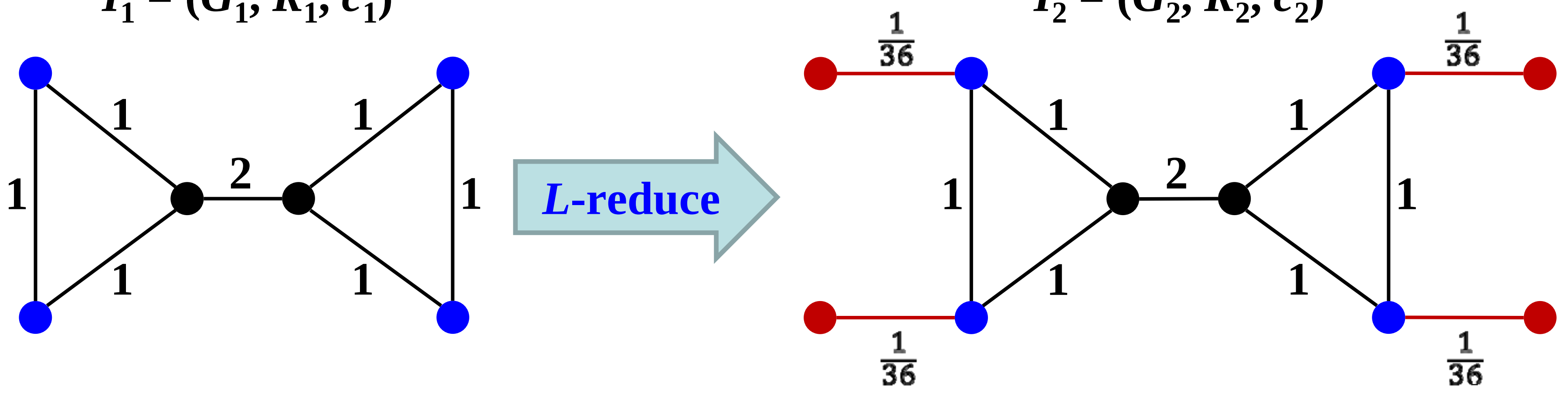
  - **Polynomial-time Algorithm A_1 produces an instance $A_1(I_1)=I_2$ for P_2 such that $\mathrm{OPT}(I_2) \le c^*\mathrm{OPT}(I_1)$**

  - **Given any solution of I_2 with cost$_2$, polynomial-time Algorithm A_2 produces a solution for I_1 of P_1 with cost$_1$ such that $|\mathrm{cost}_1 - \mathrm{OPT}(I_1)| \le d^* |\mathrm{cost}_2 - \mathrm{OPT}(I_2)|$**

MAX SNP-hardness (3/11)

- **Algorithm $A_1(I_1)$**
  - **Input: An instance $I_1 = (G_1, R_1, c_1)$ of STP(1,2), where $G_1 = (V_1, E_1)$ and $R_1 = \{r_1, r_2, \ldots, r_k\}$ for $k \geq 2$.**
  - **Output: An instance $I_2 = (G_2, R_2, c_2)$ of ISTP.**

$I_1 = (G_1, R_1, c_1)$ $I_2 = (G_2, R_2, c_2)$



● ∈ R

● ∈ dummy vertex set DV

—— ∈ dummy edge set DE

MAX SNP-hardness (4/11)

- **Lemma 2.** $\text{OPT}(I_2) \le 2\text{OPT}(I_1)$.

Pf.

T_S^*: minimum Steiner tree of I_1

$r_1, r_2, \ldots, r_q$: the leaf-terminals of T_S^*, where $q \le k$

According to Algorithm $A_1(I_1)$, there are q dummy edges incident to $r_1, r_2, \ldots, r_q$ in T_S^*, so we obtain an internal Steiner tree of $A_1(I_1)$ $(= I_2)$.

$$\Rightarrow \quad \text{OPT}(I_2) \le \text{OPT}(I_1) + \frac{q}{n^2}$$

$$\le \text{OPT}(I_1) + \frac{k}{n^2}$$

$$\le \text{OPT}(I_1) + \frac{1}{n}$$

$$\le \text{OPT}(I_1) + \frac{\text{OPT}(I_1)}{n}$$

$$= (1 + \frac{1}{n})\text{OPT}(I_1)$$

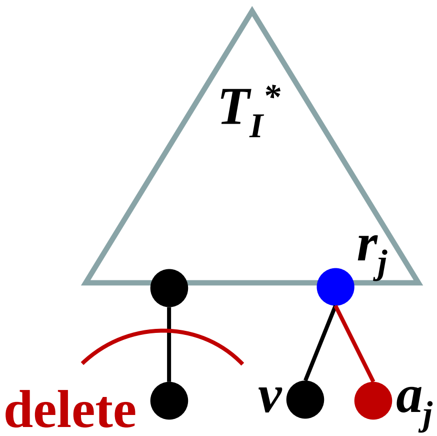
$$\le 2\text{OPT}(I_1)$$

MAX SNP-hardness (5/11)

- **Lemma 3. If T_I^* is a minimum internal Steiner tree of I_2, then T_I^* contains at least one dummy vertex, and each dummy vertex must be a leaf in T_I^*.**

***Pf*: (By contradiction)**

T_I^*: minimum internal Steiner tree of I_2; $AV(T_I^*) = \phi$

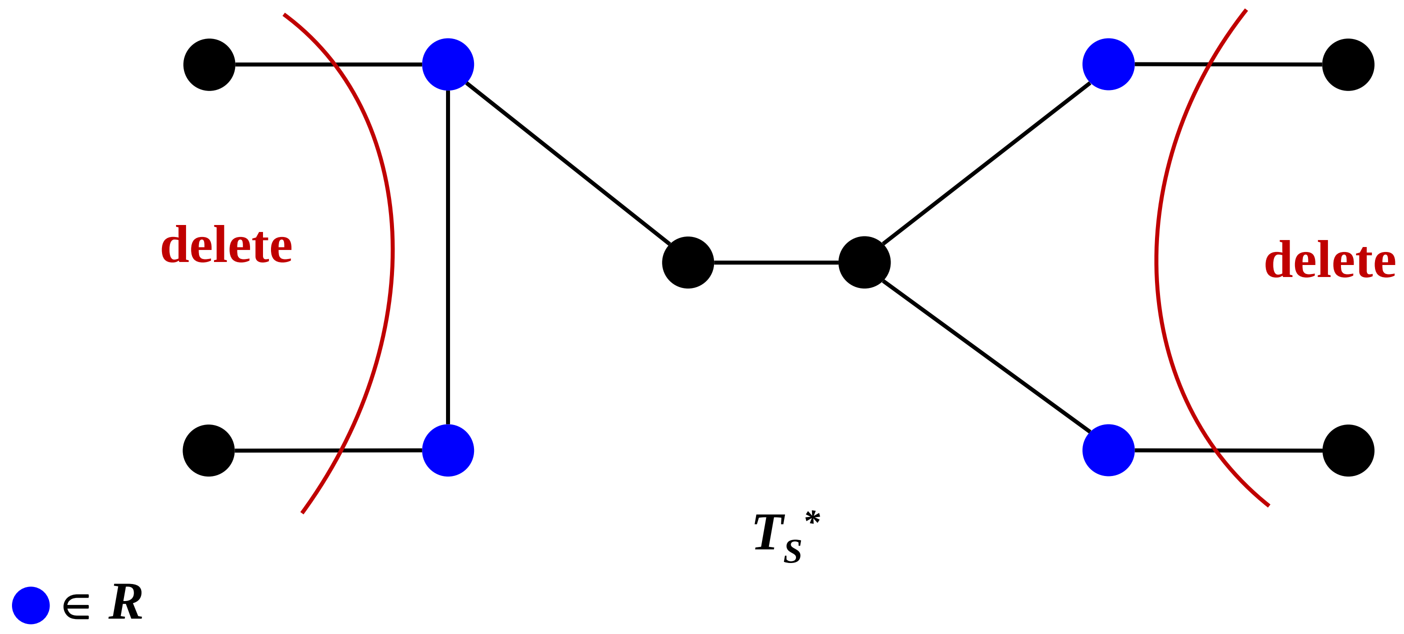


Since $c(a_j, r_j) = \frac{1}{n^2} < 1 \leq c(v, r_j)$
$(T_I^* - c(v, r_j)) + (a_j, r_j)$ is an internal Steiner tree with a lower cost $\rightarrow\leftarrow$

According to Algorithm $A_1(I_1)$, each dummy vertex must be a leaf in T_I^*.

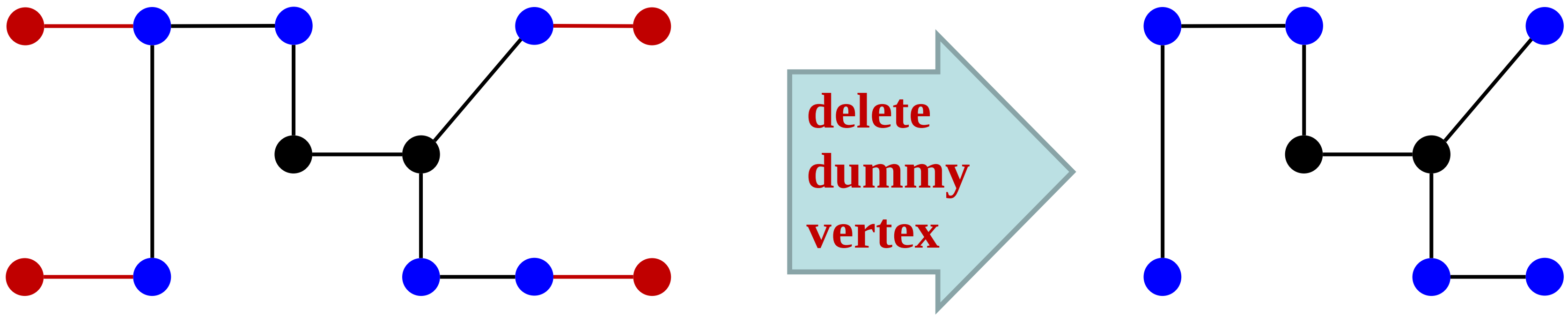
- **Lemma 4. If T_S^* is a minimum Steiner tree of I_1, then T_S^* contains at least one leaf-terminal.**

***Pf*: (By contradiction)**



MAX SNP-hardness (7/11)

- **Lemma 5. Suppose that T_I^* is a minimum internal Steiner tree of I_2. If we delete each dummy vertex from T_I^*, the resulting graph will be a minimum Steiner tree.**



T_I^* minimum internal Steiner tree **minimum Steiner tree**

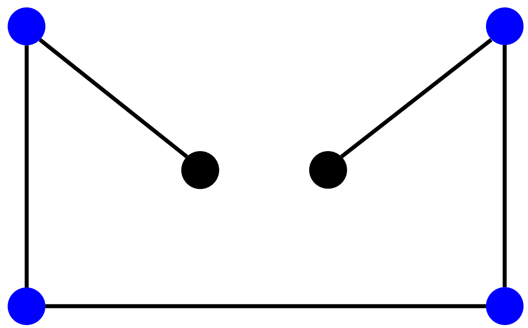
● $\in R$

● dummy vertex

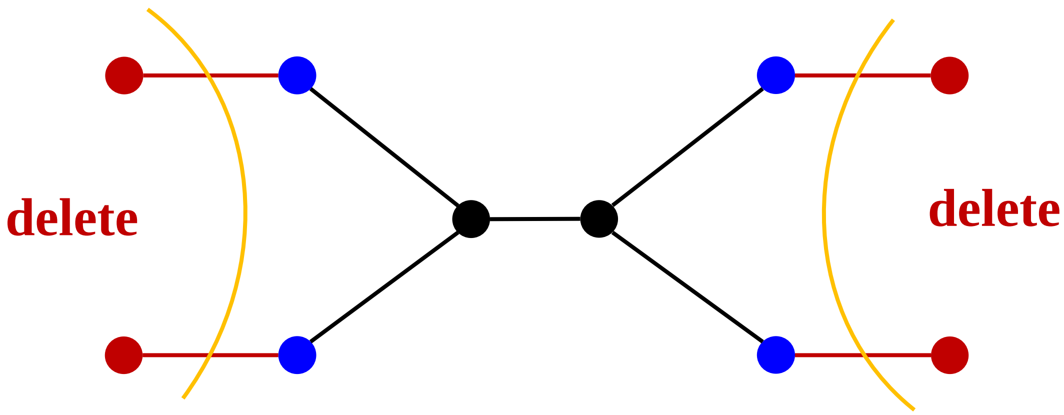
MAX SNP-hardness (8/11)

- Algorithm A_2
  - Input: An internal Steiner tree T_I of I_2=A_1(I_1), where I_1 = (G_1, R_1, c_1).
  - Output: A Steiner tree T_S of I_1.

T_I does not contain an dummy vertex



T_I contains dummy vertices



MAX SNP-hardness (9/11)

- **Lemma 6. Algorithm A_2 has the following two properties.**
  - **1) Given any internal Steiner tree T_I (a feasible solution) of $I_2 = A_1(I_1)$, the $A_2(T_I, I_2)$ algorithm will generate a Steiner tree T_S (a feasible solution) of I_1.**
  - **2) If $c(T_I) = \text{cost}_2$ and $c(T_S) = \text{cost}_1$, then $\text{cost}_1 = \text{cost}_2 - \frac{1}{n^2}|DV(T_I)|$.**

Pf.

1)

By algorithm $A_1(I_1)$, any dummy vertex must be a pendent vertex that not in R.

If T_I contains an dummy vertex, that vertex must be a leaf.

After deleting all the dummy vertices using Algorithm $A_2(T_I, I_2)$, the resulting tree must contain all vertices in R.

MAX SNP-hardness (10/11)

- **Lemma 6. Algorithm A_2 has the following two properties.**
  - 1) Given any internal Steiner tree T_I (a feasible solution) of $I_2 = A_1(I_1)$, the $A_2(T_I, I_2)$ algorithm will generate a Steiner tree T_S (a feasible solution) of I_1.
  - 2) If $c(T_I) = \text{cost}_2$ and $c(T_S) = \text{cost}_1$, then $\text{cost}_1 = \text{cost}_2 - \frac{1}{n^2}|EV(T_I)|$.

Pf.

2)

Case 1. $AV(T_I) = \phi$.

$$\text{cost}_2 = c(T_I) = \text{cost}_1 - 0$$

Case 2. $AV(T_I) \neq \phi$.

T_S is obtained from T_I by deleting $av(T_I)$ dummy edges with the cost $\frac{1}{n^2}$

$$\Rightarrow \text{cost}_1 = \text{cost}_2 - \frac{1}{n^2}av(T_I)$$

MAX SNP-hardness (11/11)

- **Theorem 1. ISTP is MAX SNP-hard.**

Pf: by providing an L-reduce from STP(1,2) to ISTP

Algorithms A_1 and A_2 are the two polynomial-time algorithm for the reduction.

By Lemma 2, $\mathrm{OPT}(A_1(I_1)) \le 2\mathrm{OPT}(I_1)$.

If $\mathrm{cost}_2 = \mathrm{OPT}(A_1(I_1))$.

According to Lemma 5, $\mathrm{cost}_1 = \mathrm{OPT}(I_1)$.

$\Rightarrow |\mathrm{cost}_1 - \mathrm{OPT}(I_1)| = 0 = |\mathrm{cost}_2 - \mathrm{OPT}(A_1(I_1))|$ and

$$\mathrm{OPT}(I_1) = \mathrm{OPT}(A_1(I_1)) - \frac{1}{n^2}|DV(T_I)|. \text{ —— } (\bigstar)$$

$$\Rightarrow |\mathrm{cost}_1 - \mathrm{OPT}(I_1)| = |\mathrm{cost}_2 - \frac{1}{n^2}|DV(T_I)| - \mathrm{OPT}(I_1)|$$

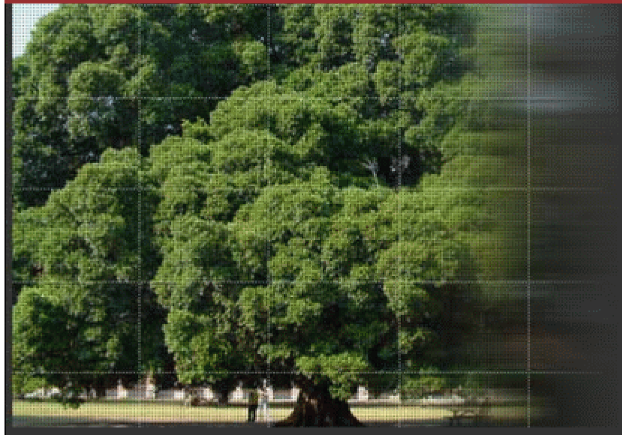
$$= |\mathrm{cost}_2 - \frac{1}{n^2}|DV(T_I)| - \mathrm{OPT}(A_1(I_1)) - \frac{1}{n^2}|DV(T_I)|| \quad \text{/* by } (\bigstar) \text{ */}$$

$$\vdots$$

$$\le |\mathrm{cost}_2 - \mathrm{OPT}(A_1(I_1))| + |\mathrm{cost}_2 - \mathrm{OPT}(A_1(I_1))|$$

$$\le 2\,|\mathrm{cost}_2 - \mathrm{OPT}(A_1(I_1))|$$

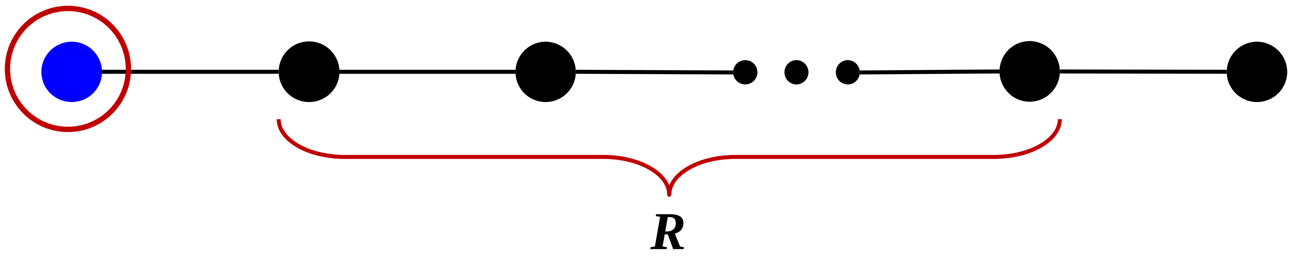
National Cheng Kung University



Approximation Algorithm

Approximation algorithm (1/11)

- **Lemma 7. Suppose that T is a Steiner tree with $|V(T) \setminus R| \geq 2$. If v is a leaf-terminal of T, then there exists an internal vertex (critical vertex) $\alpha_v \in V(T)$ such that one of the following two conditions holds:**
  - **1) $\deg_T(\alpha_v) = 2$ and $\alpha_v \notin R$; or**
  - **2) $\deg_T(\alpha_v) \geq 3$.**



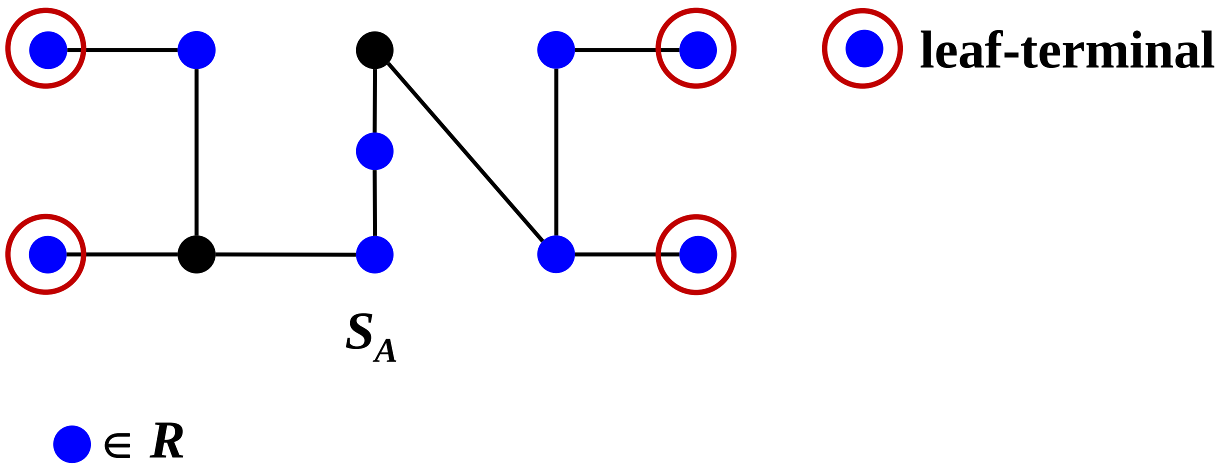
$|V(T) \setminus R| \leq 1 < 2 \rightarrow\leftarrow$

Approximation algorithm (2/11)

- **Algorithm AISTP**
  - **Input: A complete graph $G = (V, E)$ with a metric cost function $c: E \rightarrow \mathbf{R}^+$ and a proper subset $R \subset V$ of terminals such that $|V \backslash R| \geq 2$.**
  - **Output: An internal Steiner tree T_I.**

Step 1. Find a Steiner tree S_A in G

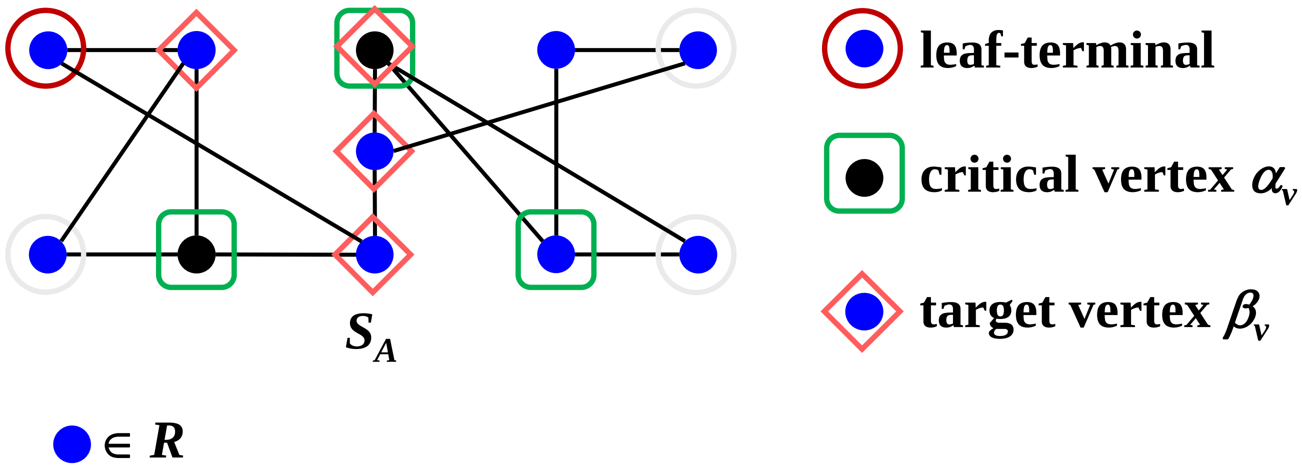
Use the currently best-known approximation algorithm



Approximation algorithm (3/11)

- **Algorithm AISTP**
  - **Input: A complete graph $G = (V, E)$ with a metric cost function $c: E \rightarrow \mathbf{R}^+$ and a proper subset $R \subset V$ of terminals such that $|V \backslash R| \geq 2$.**
  - **Output: An internal Steiner tree T_I.**

Step 2. Transform T into an internal Steiner tree



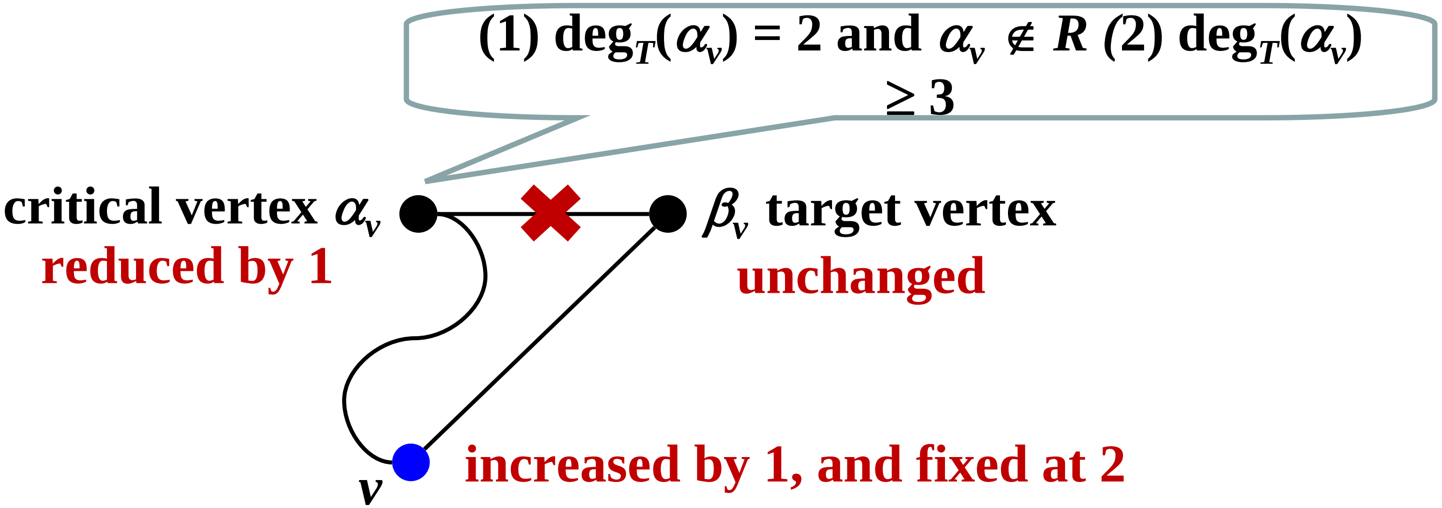
Approximation algorithm (4/11)

- **Lemma 8.**

  v: a leaf-terminal of the current tree T selected for processing. Then,

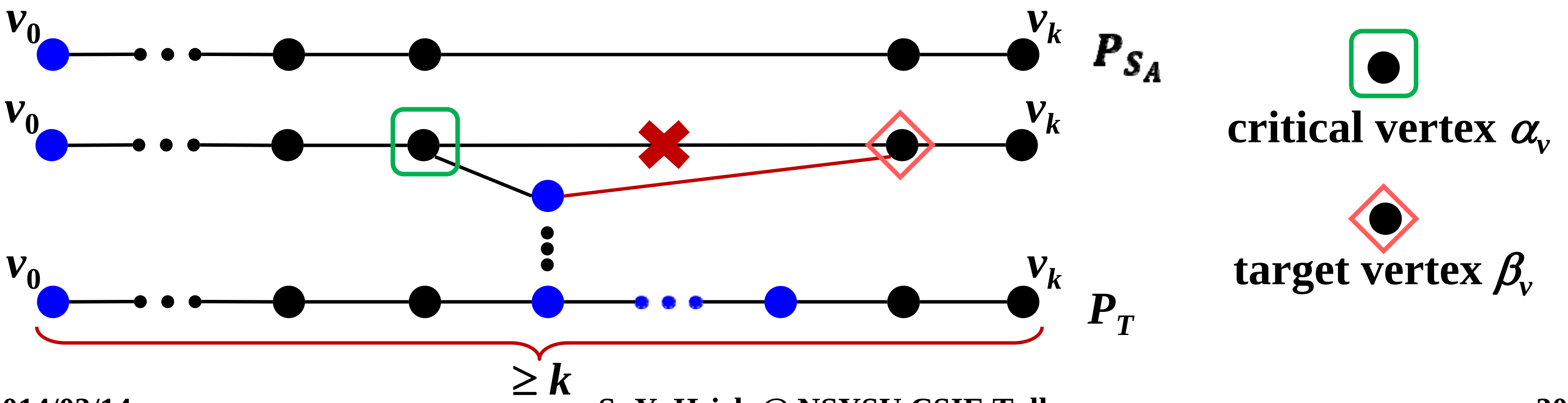
  - **1) $\deg_T(\alpha_v)$ will be reduced by 1 after the iteration.**
  - **2) $\deg_T(\beta_v)$ will be unchanged after the iteration.**
  - **3) $\deg_T(v)$ will be increased by 1, and fixed at 2 until the algorithm terminates.**

***Pf*:**



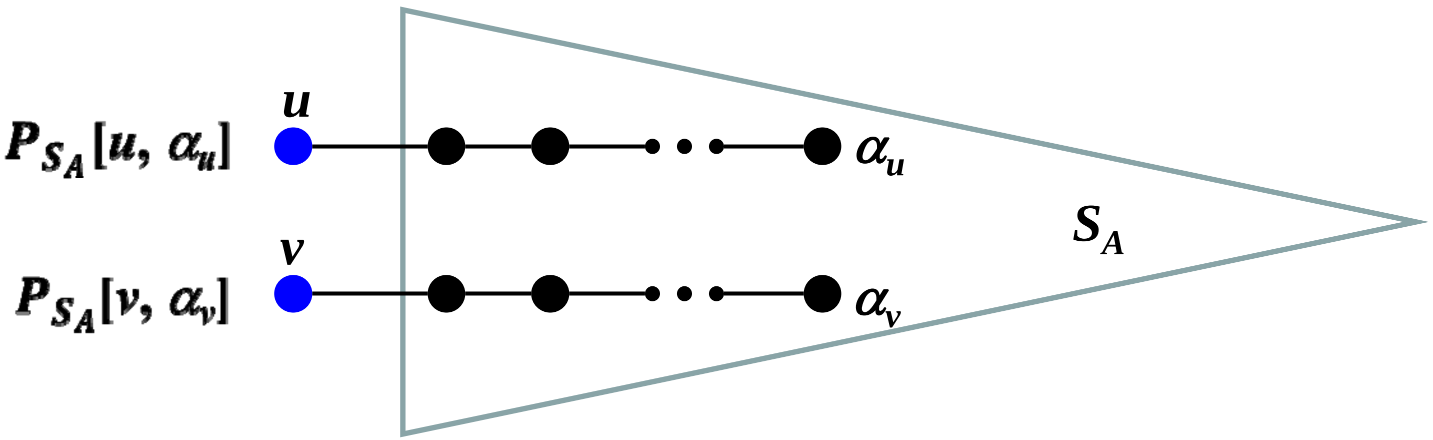
Approximation algorithm (5/11)

- **Lemma 9.** Suppose that v_0 is a leaf-terminal of S_A, and v_k is the critical vertex of v_0 selected by Algorithm A_{ISTP}, i.e., $\alpha_{v_0} = v_k$. Let $P_{S_A}[v_0, v_k] = \langle v_0, v_1, \ldots, v_k \rangle$ be a path of S_A. Then, during the execution of Algorithm A_{ISTP}, before v_0 is processed, there always exists a path $P_T[v_0, v_k]$ in the current tree T, which is extended from $P_{S_A}[v_0, v_k]$ such that the following properties hold:
  - 1) The length of $P_T[v_0, v_k]$ is at least k.
  - 2) If $V(P_T[v_0, v_k]) \setminus V(P_{S_A}[v_0, v_k]) \neq \phi$, then the vertices in $V(P_T[v_0, v_k]) \setminus V(P_{S_A}[v_0, v_k])$ are all in R, and each vertex in $V(P_T[v_0, v_k]) \setminus V(P_{S_A}[v_0, v_k])$ has a fixed degree of 2 until the algorithm terminates.



Approximation algorithm (6/11)

- **Lemma 10. Suppose that u and v are two leaf-terminals of S_A such that u is selected before v for processing by the A_{ISTP} algorithm. Then, the two path $P_{S_A}[u, \alpha_u]$ and $P_{S_A}[v, \alpha_v]$ are edge-disjoint.**

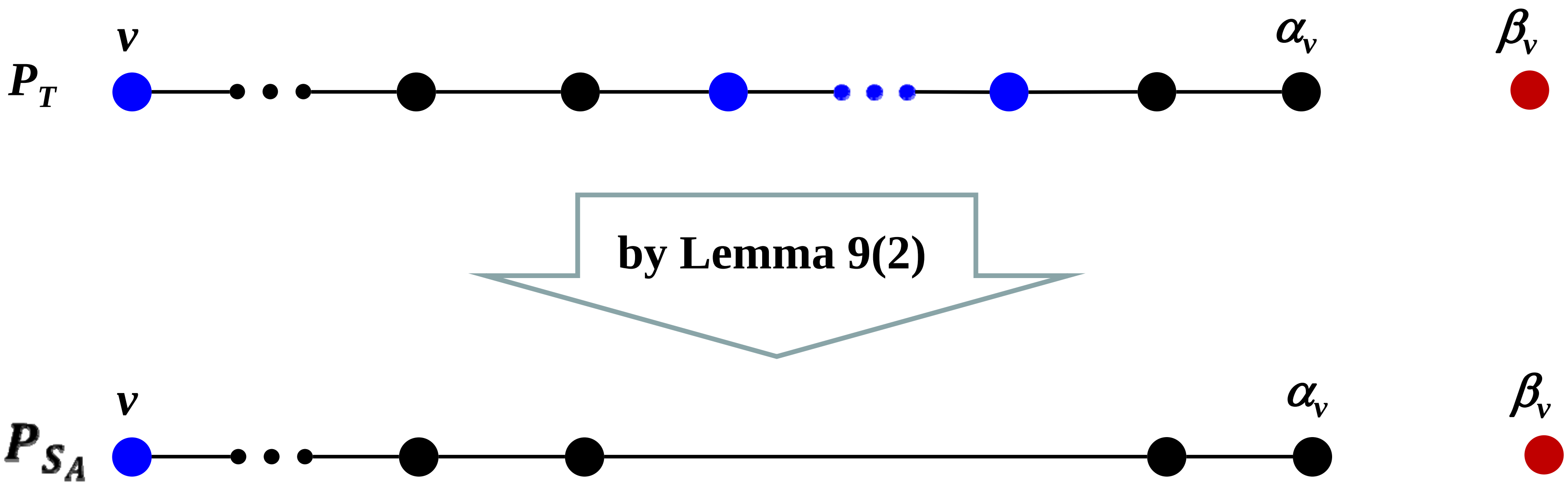


- **Lemma 11. Let $v_1, v_2, \ldots, v_l$ be the order of the leaf-terminals of S_A selected for processing by the A_{ISTP} algorithm. Then, the paths $P_{S_A}[v_1, \alpha_{v_1}]$, $P_{S_A}[v_2, \alpha_{v_2}]$,…, $P_{S_A}[v_l, \alpha_{v_l}]$, are pairwise edge-disjoint.**

Approximation algorithm (7/11)

- **Lemma 12.** Suppose that v is a leaf-terminal of S_A. Then, the target vertex β_v cannot belong to $P_{S_A}[v, \alpha_v]$.

according to step 2(b), $\beta_v \notin P_T[v, \alpha_v]$



Approximation algorithm (8/11)

- **Lemma 13. Let $P = \langle v_1, v_2, \ldots, v_{k-1}, v_k \rangle$ be a path of a graph $G = (V, E)$ with a metric cost function c: $E \to \mathbb{R}^+$, and let $P' = \langle v_1, v_2, \ldots, v_{k-2}, v_{k-1} \rangle$ be a subpath of P. Then, $c(v_1, v_k) - c(v_{k-1}, v_k) \le c(P')$, where $c(P') = \sum_{j=1}^{k-2} c(v_j, v_{j+1})$.**

***Pf*:**

Since c is metric,

$c(v_1, v_k) - c(v_{k-1}, v_k) \le c(v_1, v_{k-1})$

$c(v_1, v_{k-1}) - c(v_{k-2}, v_{k-1}) \le c(v_1, v_{k-2})$

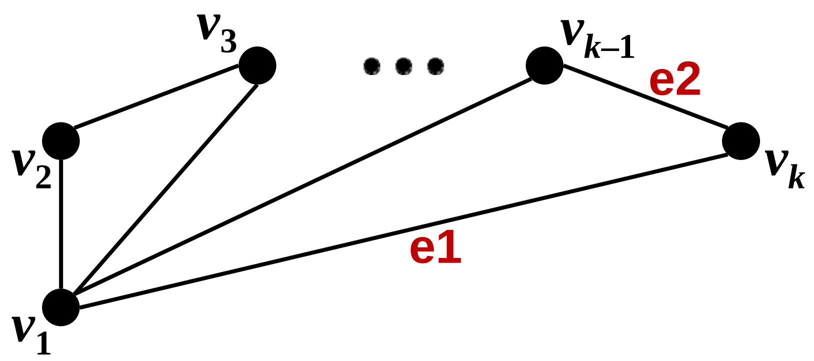
$c(v_1, v_{k-2}) - c(v_{k-3}, v_{k-2}) \le c(v_1, v_{k-3})$

$\vdots$

$c(v_1, v_4) - c(v_3, v_4) \le c(v_1, v_3)$, and

$c(v_1, v_3) - c(v_2, v_3) \le c(v_1, v_2)$

$\Rightarrow c(v_1, v_k) - c(v_{k-1}, v_k) \le c(v_1, v_2) + c(v_2, v_3) + \ldots + c(v_{k-3}, v_{k-2}) + c(v_{k-2}, v_{k-1}) = c(P')$



Approximation algorithm (9/11)

- **Theorem 2. Algorithm A_{ISTP} is a $(2\rho + 1)$-approximation algorithm for the internal Steiner tree problem on complete graphs, where ρ is the approximation ratio of the best-known algorithm for the Steiner tree problem.**

***Pf*: T_I^*: minimum internal Steiner tree T_S^*: minimum Steiner tree**

Use a ρ-approximation algorithm to find a Steiner tree S_A and transform it into an internal Steiner tree.

$$c(S_A) \le \rho c(T_S^*)$$

Since T_I^* is also a Steiner tree for R.

$$c(T_S^*) \le c(T_I^*)$$

$$\Rightarrow c(S_A) \le \rho c(T_I^*) \quad \text{———} \quad (\bigstar)$$

Approximation algorithm (10/11)

- **Theorem 2. Algorithm A_{ISTP} is a $(2\rho + 1)$-approximation algorithm for the internal Steiner tree problem on complete graphs, where ρ is the approximation ratio of the best-known algorithm for the Steiner tree problem.**

***Pf*:**

$$
\begin{aligned}
c(T_I) &\le c(S_A) + \Sigma_{v\in R}(c(v, \beta_v) - c(\alpha_v, \beta_v)) + \mu \\
&\le c(S_A) + \Sigma_{v\in R}(c(P_{S_A}[v, \alpha_v])) + \mu \\
&\le c(S_A) + c(S_A) + \mu \\
&= 2c(S_A) + \mu \\
&\le 2\rho c(T_I^*) + \mu \qquad \text{/* by } (\bigstar) \text{ */}
\end{aligned}
$$

$$
\Rightarrow \frac{c(T_I)}{c(T_I^*)} \le 2\rho + \frac{\mu}{c(T_I^*)} = 2\rho + 1
$$

Approximation algorithm (11/11)

- **Theorem 3. Algorithm A_{ISTP} can be implemented to run in $O(n^2\log n) + f(m, n)$ time, where $f(n, m)$ is the time complexity of the best-known approximation algorithm for the Steiner tree problem in an input graph with n vertices and m edges.**

Pf: n: number of vertices; m: number of edges

Step 1. $f(n, m)$: the best-known approximation algorithm

Step 2. $O(m\log n)$: adding the needed vertices and edges by sorting the cost

$O(|V(T)+|E(T)|)$: select the nearest vertex αv using a depth-first search

$\Rightarrow O(|R|(|V(T)|+|E(T)|) = O(n^2)$

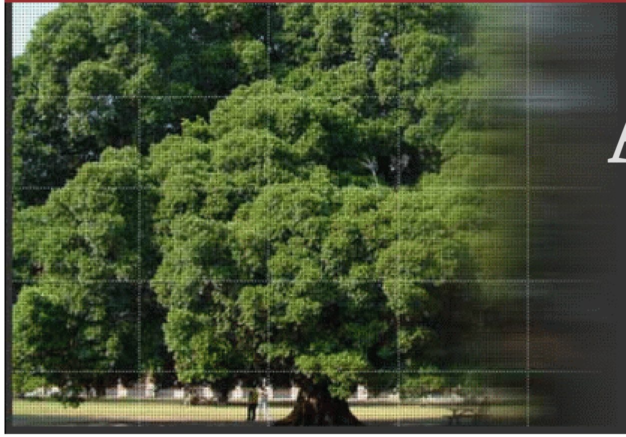
$O(\deg_T(v))$: choose a vertex βv

$\Rightarrow O(\Sigma_{v\in R}\deg_T(v)) = O(|E(T)|) = O(n)$

$\Rightarrow \quad O(m\log n) + O(n^2) + f(n, m)$

$= O(n^2\log n) + f(n, m)$ since an input graph is a complete graph

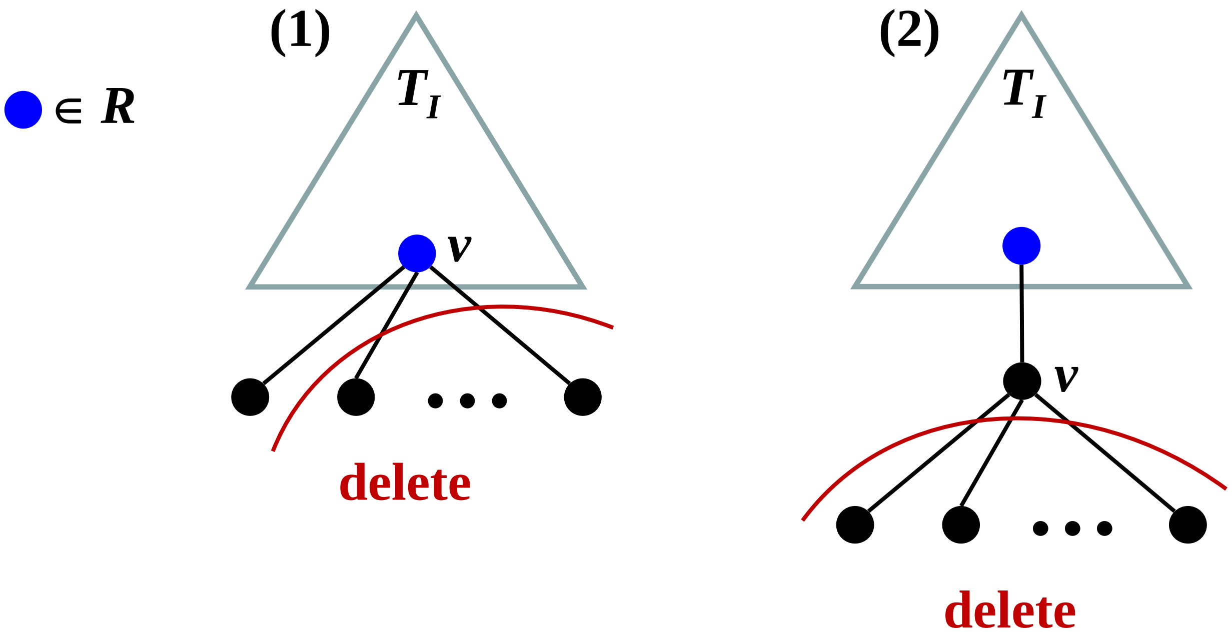
National Cheng Kung University



Approximation Algorithm for ISTP(1,2)

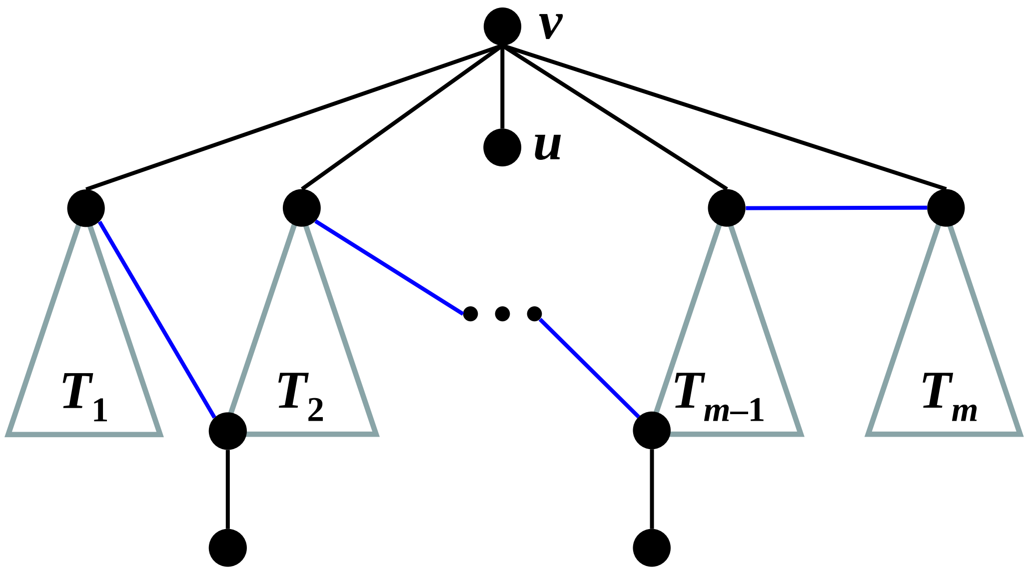
ISTP(1,2) (1/7)

- We restrict the co-domain of the cost function to the range $\mathbf{R}^+$ to {1, 2}.
- **Lemma 14.** Let *T* be a minimum (1,2)-internal Steiner tree and let *v* be an internal vertex in *T*. Then the following statements hold.
  - 1) If *v* is a terminal (i.e., $v \in R$), then at most one leaf will be adjacent it.
  - 2) If *v* is a Steiner vertex, then no leaf will be adjacent it.



ISTP(1,2) (2/7)

- **Lemma 15. Let T be a minimum (1,2)-internal Steiner tree and let v be an internal vertex in T. If v is a Steiner vertex, then there exists a minimum (1,2)-internal Steiner tree T' that does not contain v.**



$|T_i| \geq 2$

T_i contains at least one leaf

$u \in R$

$\Rightarrow$ not an internal vertex

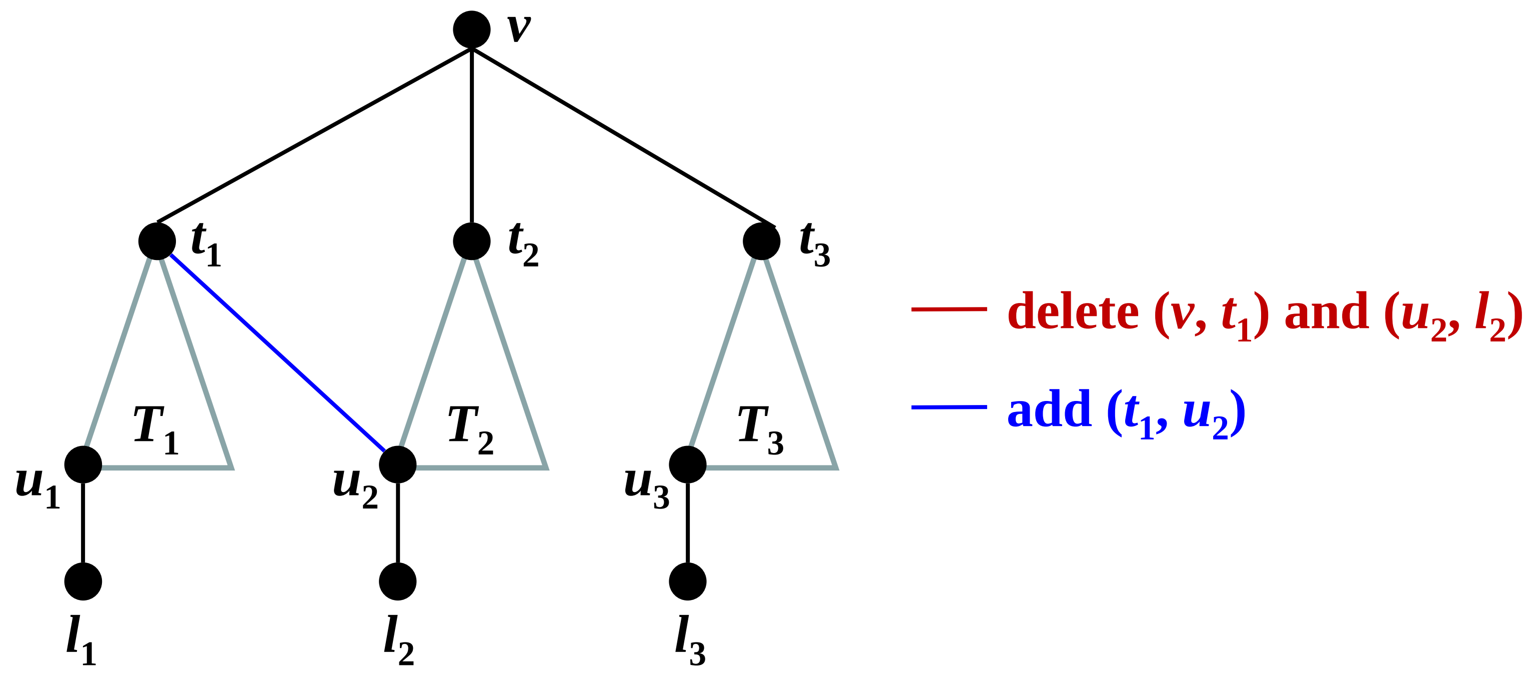
u is Steiner vertex

$\Rightarrow$ delete u, obtain another lower cost internal Steiner tree

—— delete $(m - 2) + m$ edges

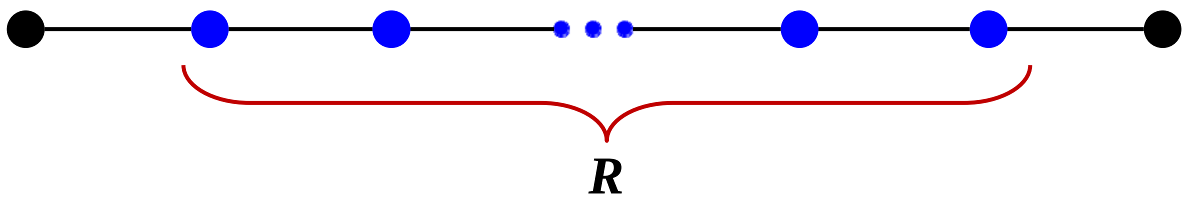
—— add $(m - 1)$ edges

- **Lemma 16. If T is a minimum (1,2)-internal Steiner tree with $k \geq 3$ leaves, then there exists a minimum (1,2)-internal Steiner tree T' with $k - 1$ leaves.**



ISTP(1,2) (4/7)

- **Lemma 17. There is an optimal solution for ISTP(1,2), which is a minimum cost path whose internal vertices are all terminals in *G* and whose end-vertices are two Steiner vertices in *G*.**



- r_i, $r_j \in R$,
  u, $v \in$ Steiner vertices,
  let $G' = G[R \cup \{u, v\}]$

- **Lemma 18. Let T^* be a minimum (1,2)-internal Steiner tree of G. Then $c(T^*) \geq c(HC_{G'}) - 2$.**

Pf:

$$c(T^*) = c(HP_{G'}[u, v])$$
$$\geq c(HC_{G'}) - c(u, v)$$
$$\geq c(HC_{G'}) - 2$$

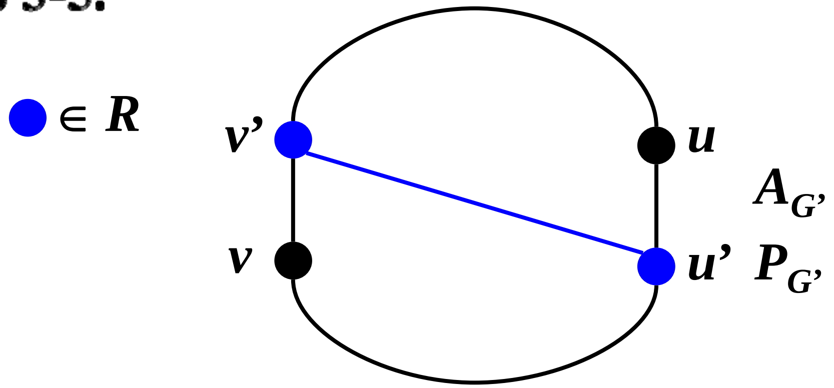
ISTP(1,2) (6/7)

- **Algorithm AISTP(1,2)**
  - **Input: A complete graph $G = (V, E)$ with a metric cost function $c: E \rightarrow \{1,2\}$ and a proper subset $R = \{r_1, r_2, \ldots, r_k\} \subset V$ of terminals such that $|V \backslash R| \geq 2$.**
  - **Output: A (1,2)-internal Steiner tree.**

Step 1. select two Steiner vertices u and v from $V \backslash R$

Step 2. use the best-known $\frac{8}{7}$-approximation algorithm for the TSP to get a Hamiltonian cycle $A_{G'}$ in G'

Step 3-5.



Lemma 19. $c(P_{G'}) \leq c(A_{G'})$

$$c(P_{G'})$$
$$= c(A_{G'}) - c(u, u') - c(v, v') + c(u', v')$$
$$\leq c(A_{G'})$$

ISTP(1,2) (7/7)

- **Theorem 4. Algorithm $A_{ISTP(1,2)}$ is a $\frac{9}{7}$-approximation algorithm for ISTP(1,2).**

Pf.

$$\frac{c(P_{G'})}{c(T^*)} \le \frac{c(A_{G'})}{c(HC_{G'})-2}$$

$$\le \frac{\frac{8}{7}c(HC_{G'})}{c(HC_{G'})-2}$$ Let n' be the number of vertices in G', $3 \le n' \le c(HC_{G'})$

$$\le \frac{8}{7} \times \frac{n'}{n'-2}$$

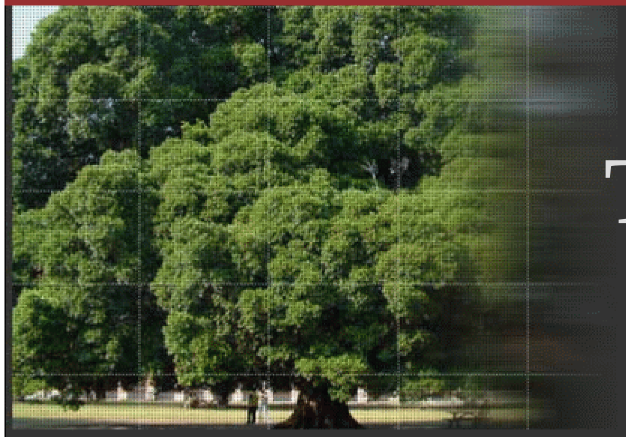
$$\le \frac{9}{7} \text{ for } n' \ge 18$$

- **Time complexity:**
  - dominated by the best-known $\frac{8}{7}$-approximation algorithm: polynomial time

Concluding Remarks

- The internal Steiner tree problem is MAX SNP-hard.
- We present a $(2\rho + 1)$-approximation algorithm for solving the problem under the metric space.
- For the case where the cost of each edge is restricted to being either 1 or 2, we propose a $\frac{9}{7}$-approximation algorithm for ISTP(1,2).
- Improving the approximation ratio of the classical Steiner tree problem.

National Cheng Kung University



Thank you for your attention!