生成树与斯坦纳树

主讲人：赵功名　　邮箱：gmzhao@ustc.edu.cn
课题组主页：http://staff.ustc.edu.cn/~xuhongli/int/
计算机科学与技术学院　　2022年10月13日

课题组
微信公众号

生成树

- **问题**（书本P39）：假设要建造一个连接n个城市v_1, v_2,···, v_n的铁路网。已知城市v_i和 v_j之间直接线路的造价是c_{ij}，如何设计一个总造价最小并且连通的铁路网？

- **答**：将其转化为边权图G，然后利用最小生成树算法构造最小生成树。

生成树

- 需要建造铁路连接城市徐州、武汉、杭州三市，求最小代价构造方案。

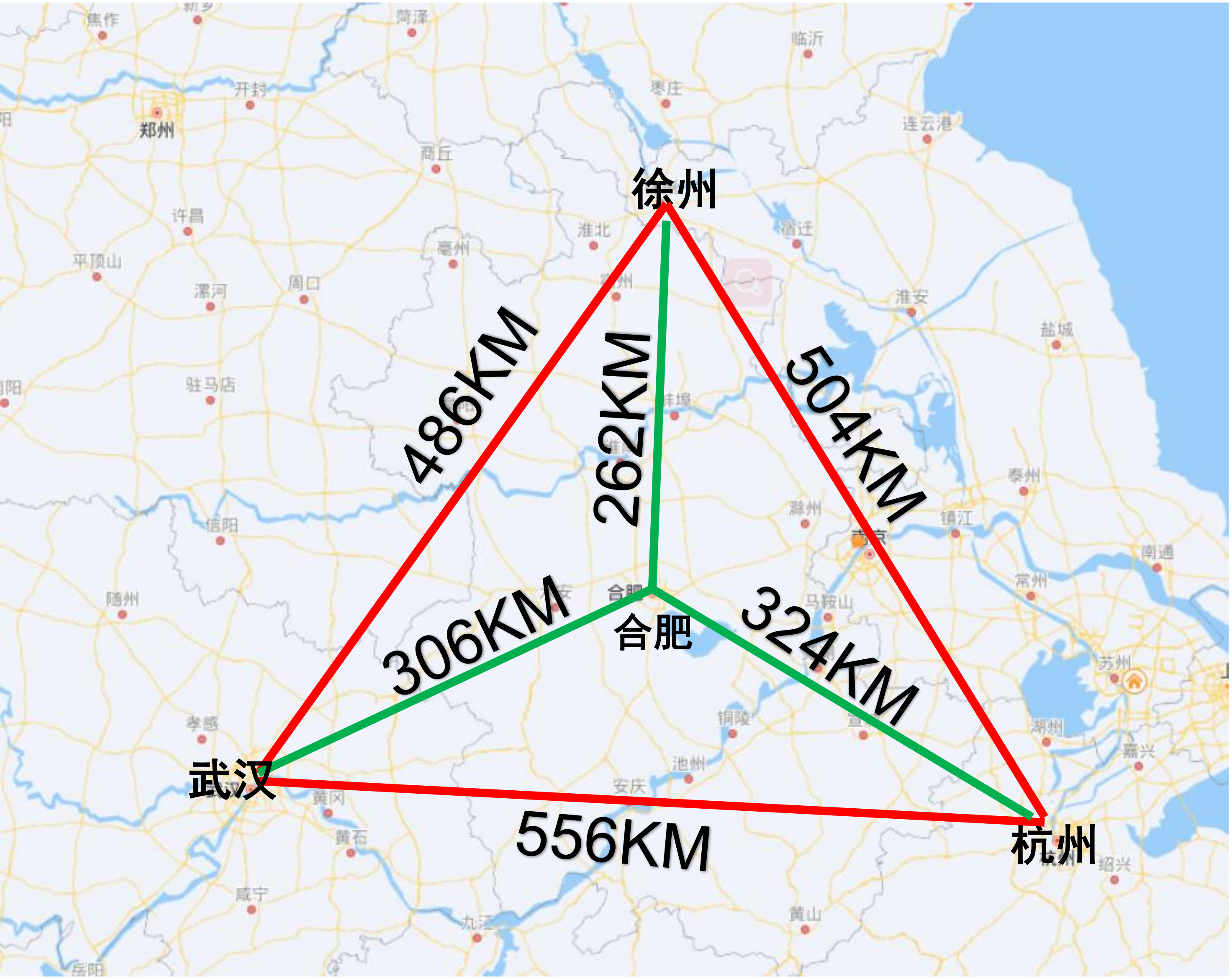
生成树

- 将其转化为边权图G，然后利用最小生成树算法构造最小生成树。

生成树

- 构造最小生成树，则建造铁路总长度为：486+504=990KM

斯坦纳树

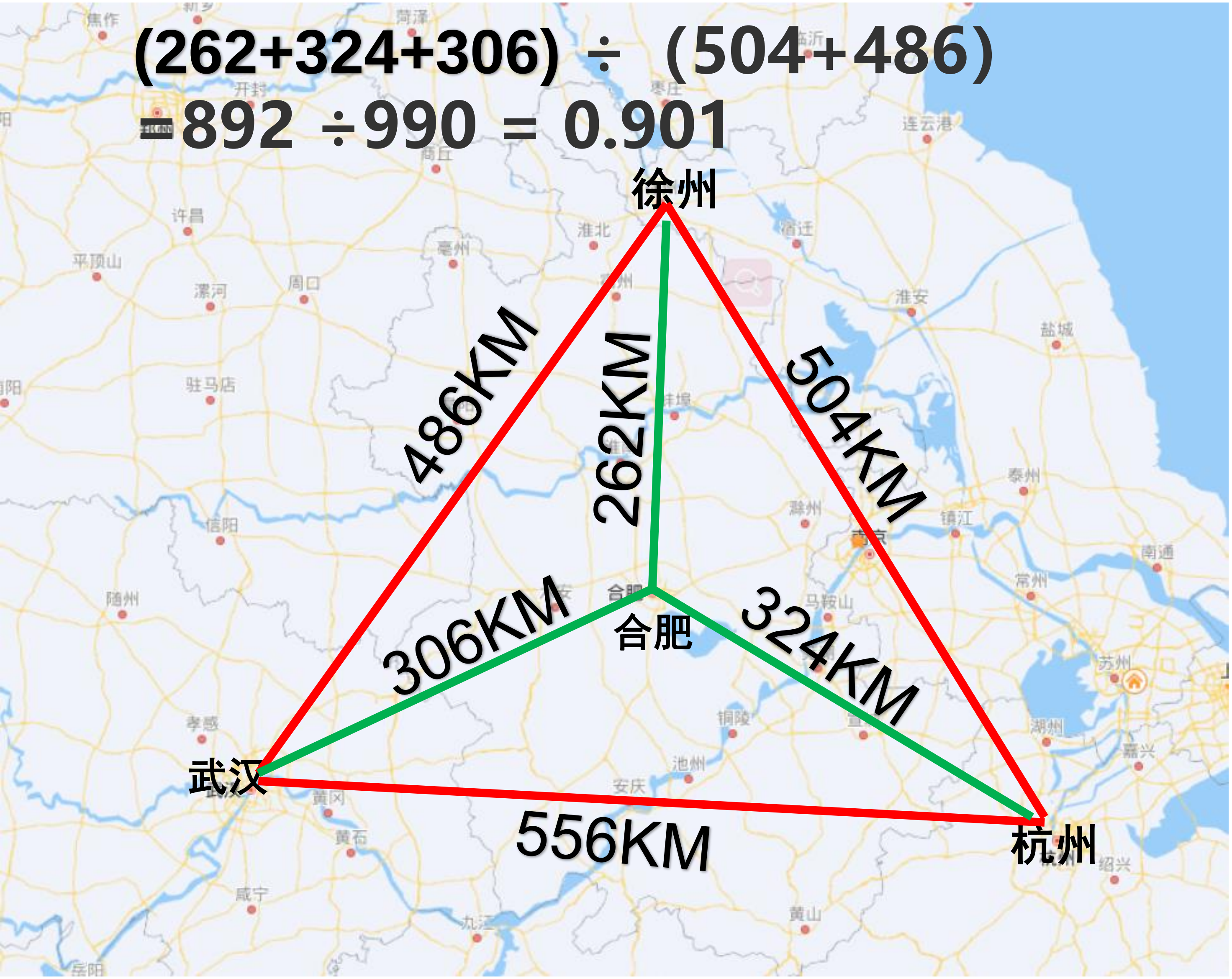


- 如果我们新找一个中转站，建造铁路来连接三个城市，能否降低成本呢？

斯坦纳树

- 以合肥为中转站，则建造铁路总长度为：262+324+306 = 892KM

斯坦纳树

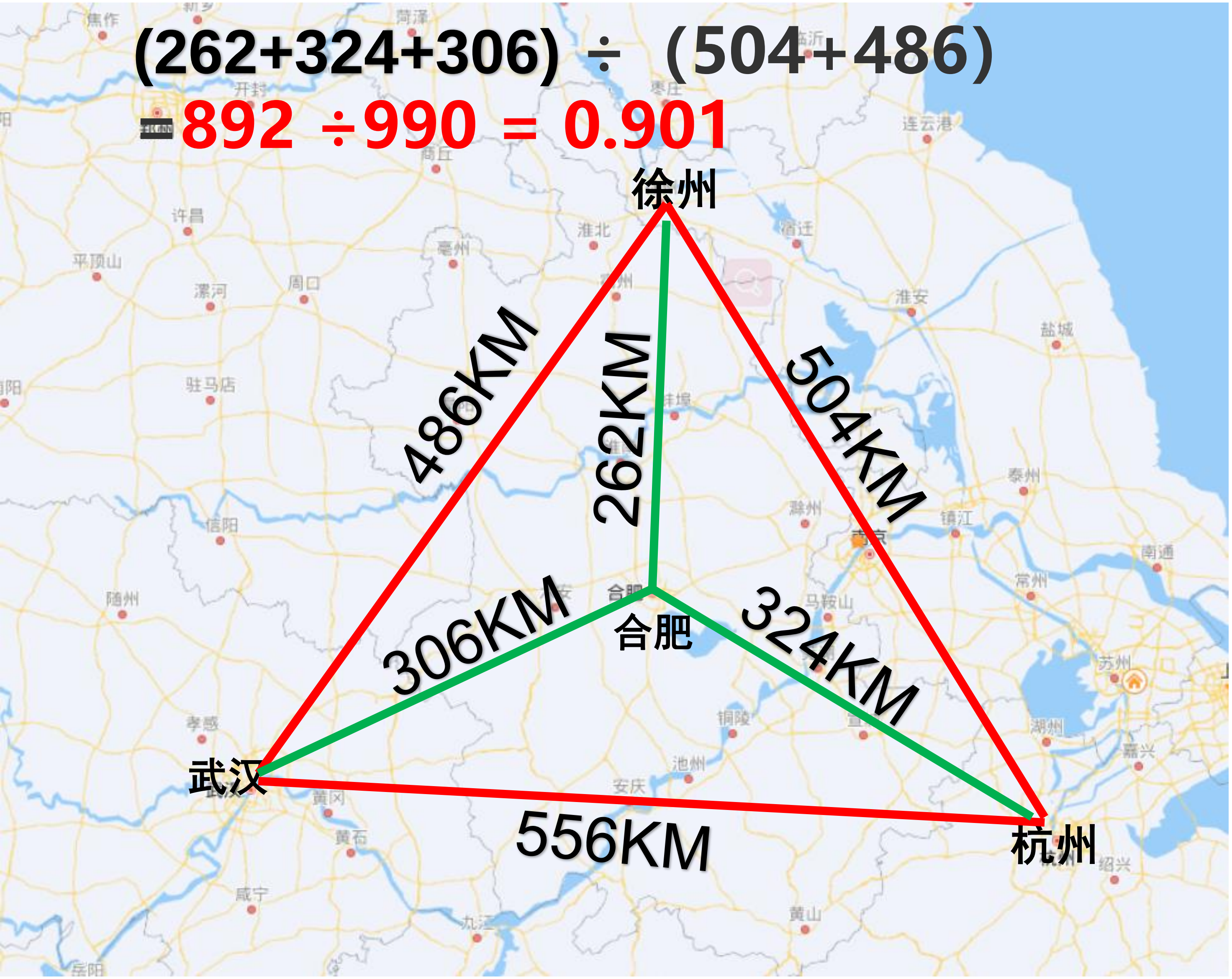


- 通过中转站合肥建造铁路连接徐州、武汉和杭州三市，可以缩短铁路建造里程。

斯坦纳树

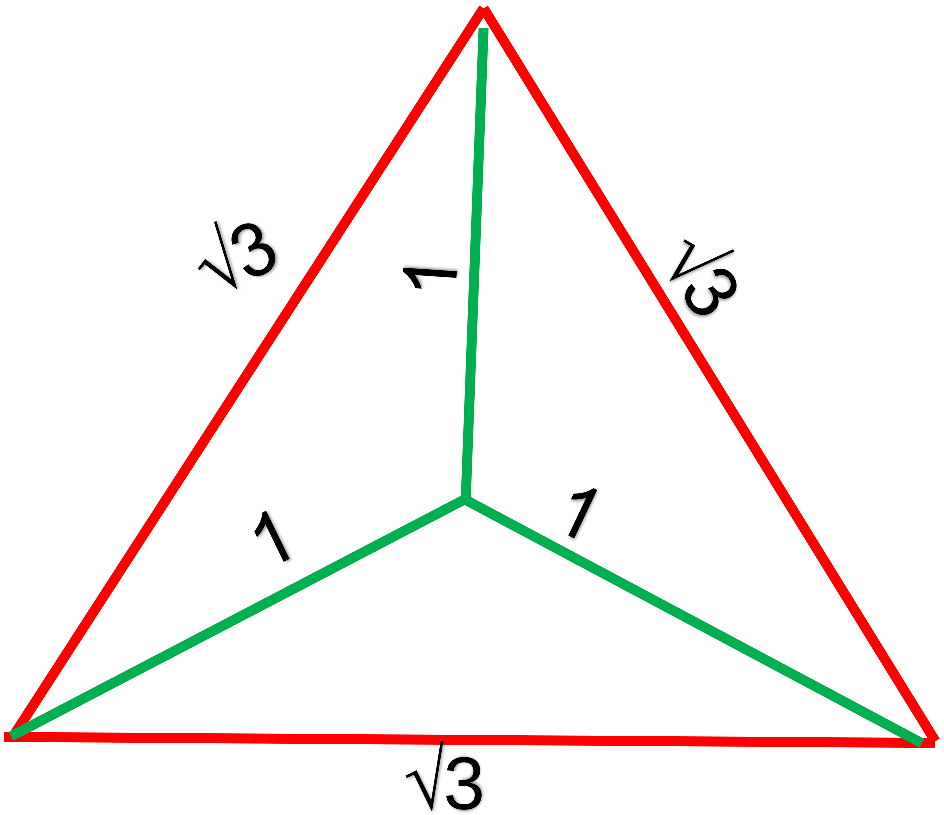
- **斯坦纳树（Steiner Tree）问题**：在一个度量空间中，给定一个点集X，通常称为**结点**集，寻找一个距离最短的连接X中的所有点的网络。由于树是一个极小的连通图，因此这样一个最小的网络一定是一棵树。我们称这样的一个最小网络为**斯坦纳最小树（SMT）**，同时称任意一个极小的连接结点集的网络为斯坦纳树。当度量空间为欧氏平面时，称这样的问题为**欧氏平面上的斯坦纳树问题**。

斯坦纳树



- 通过中转站合肥建造铁路连接徐州、武汉和杭州三市，可以缩短铁路建造里程。

吉尔伯特--波雷克猜想（斯坦纳比难题）



$3 \div 2\sqrt{3} = \sqrt{3}/2 \approx 0.866$

- 1968年贝尔实验室波雷克和吉尔伯特提出如下猜想：平面上任意点集，斯坦纳最小树长与最小生成树长的比值的最小值是$\sqrt{3}/2$（斯坦纳比）。
- 在边长为$\sqrt{3}$的等边三角形中，最小生成树的长度为$2\sqrt{3}$，而斯坦纳最小树的长度为3。

吉尔伯特--波雷克猜想（斯坦纳比难题）

- **戏剧性起源：**1967年前，美国贝尔电话公司（AT&T前身）按照连接各分部的最小生成树的长度来收费。当时一家精明的用户航空公司，要求贝尔在第四个点的位置上架上电话线。这样使得电话公司不仅要拉新线，增加服务网点，而且要减少收费。这件事的连锁反应迫使电话公司改变了坚持长达10年按照最小生成树长度收费的原则，并且不得不对最短网络问题进行研究。

网络斯坦纳树问题

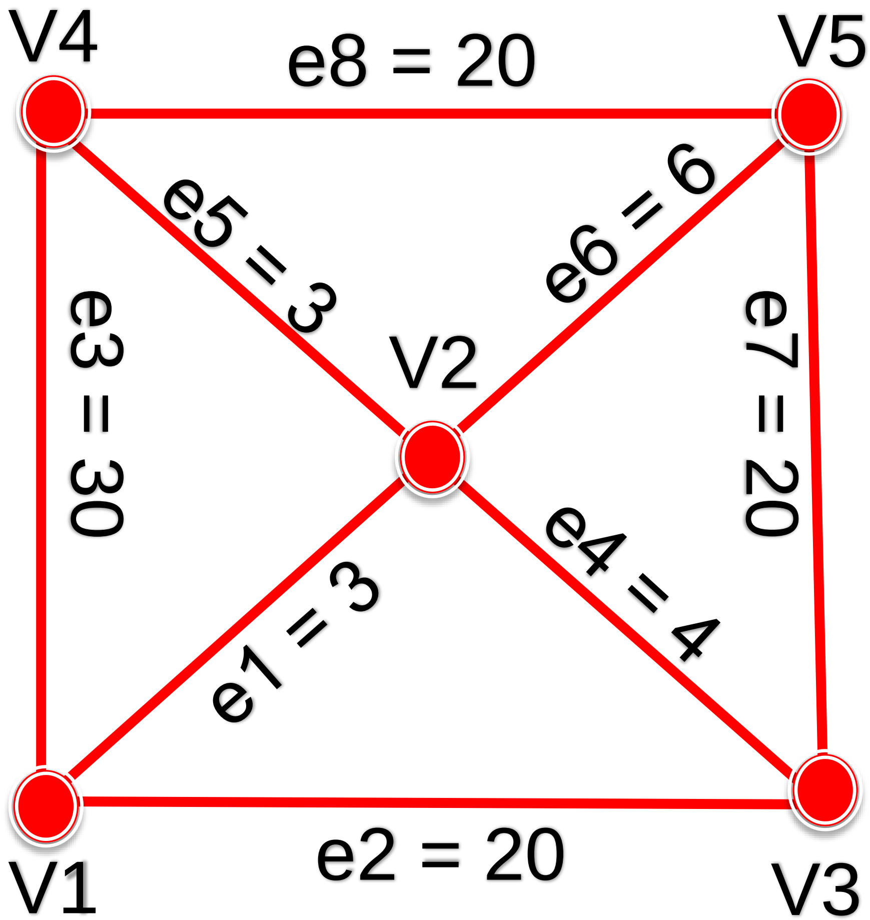
- 假设有7个城市已经建好火车站，现在需要建造最短的铁路保证徐州、武汉、杭州三市的连通。即图G包含7个点，要求保证其中给定3个点的连通性。

网络斯坦纳树问题

- **网络斯坦纳树问题**：给定一个边赋权图G=（V,E）和一个结点集X ⊆ V，图上的斯坦纳树问题是指寻找一棵权重最小且连接结点集中的所有顶点的树。
- 注：在这个问题中，通常可以假定给定的图是一个完全图，而且定义在边集上的权函数满足三角不等式。
- 理由：如果给定的图不是完全图，我们可以构造一个具有相同顶点集的完全图，在这个图中每条边(u,v)的权定义为原图u和v之间最短路的长度；如果权函数不满足三角不等式，那么可以用最短路的权重来替换原来边的权重。可以验证，这个新的满足三角不等式的完全图上的网络斯坦纳树问题与原图问题是等价的。

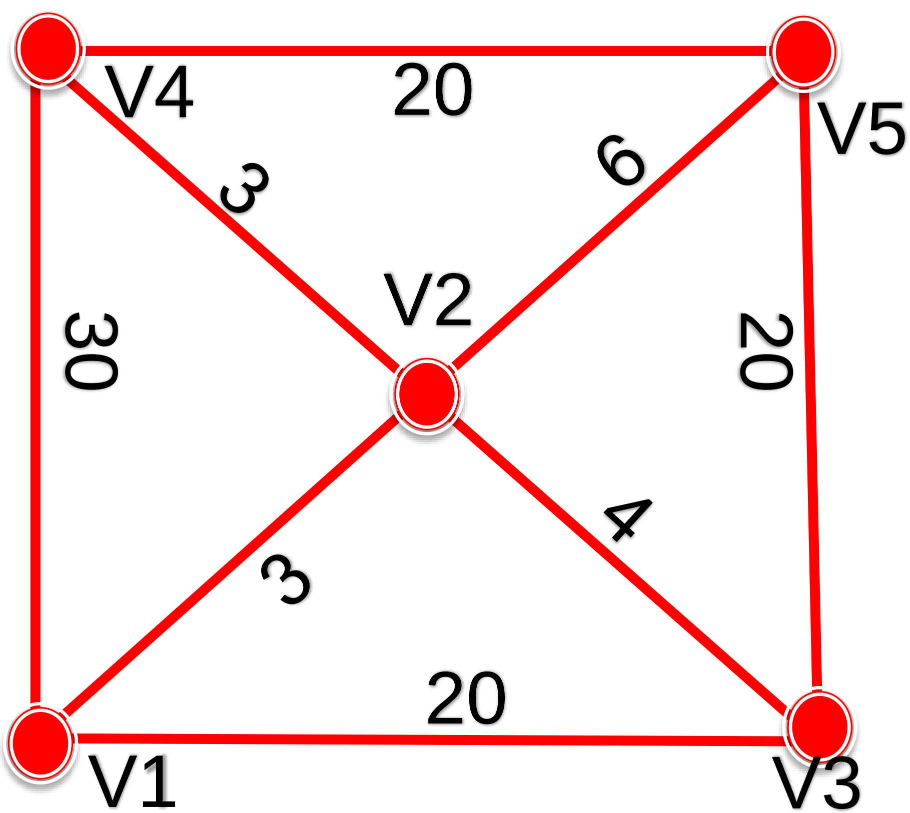
最小生成树算法求解网络斯坦纳树问题

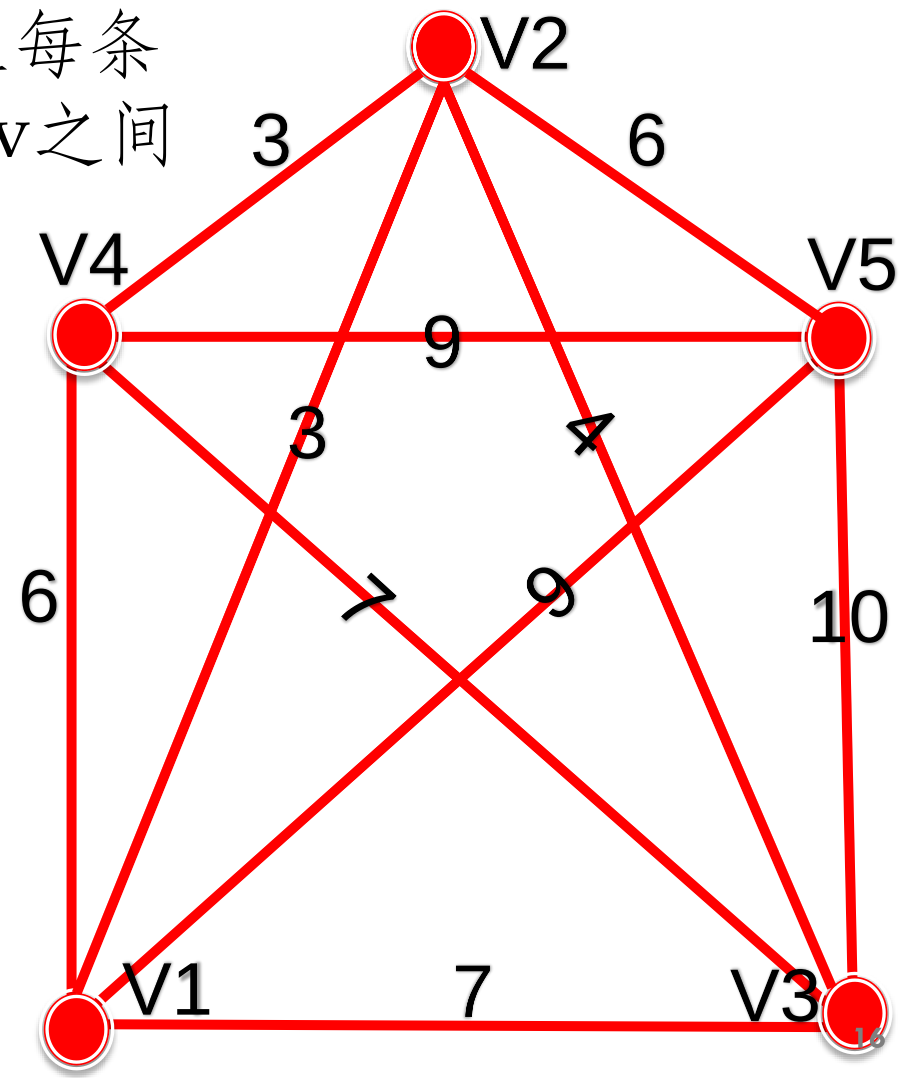
- 例题一：原始输入图G包含5个点和8条边，求包含结点集合P={V1, V3, V4}的斯坦纳树。



最小生成树算法求解网络斯坦纳树问题

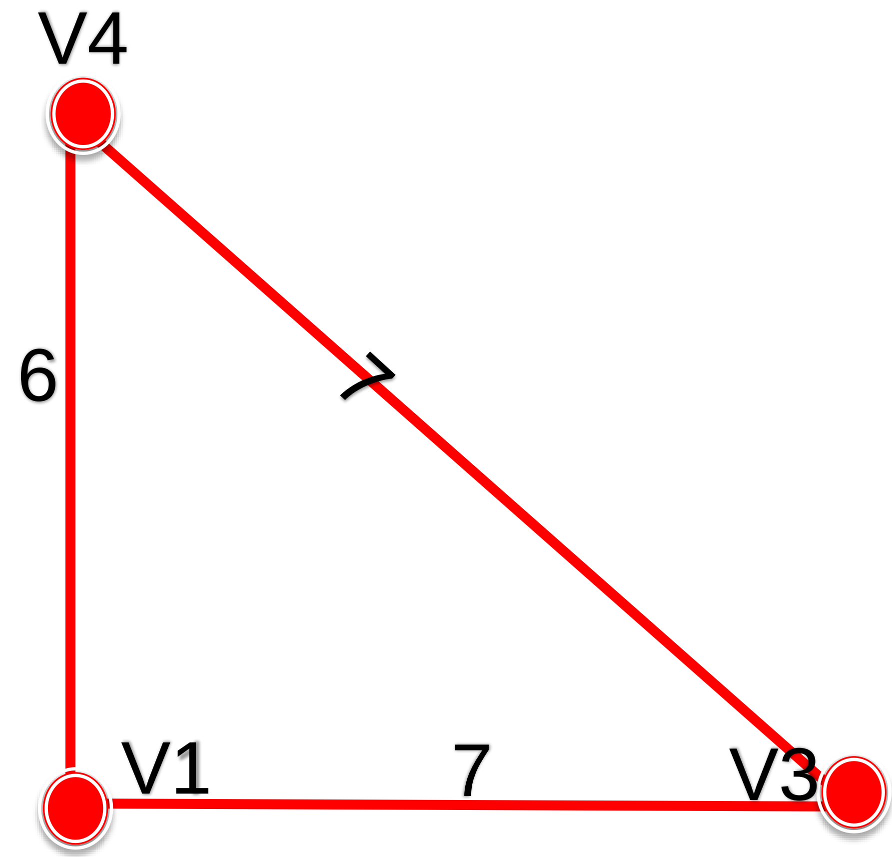
- 1、构造满足三角不等式的完全图：将其转换为完全图，且每条边(u,v)的权定义为原图u和v之间最短路的长度。





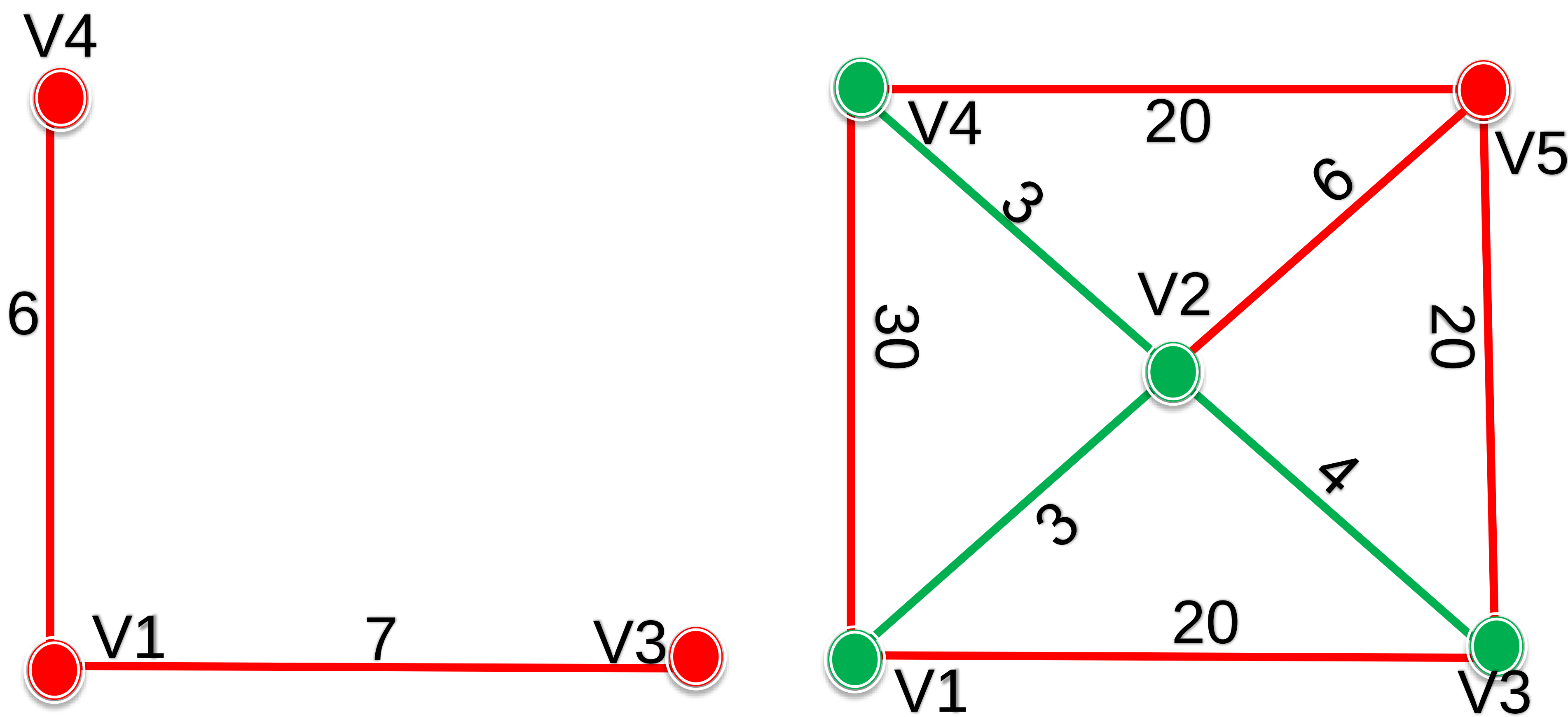
最小生成树算法求解网络斯坦纳树问题

- 2、将完全图转化为一个只包含斯坦纳点的完全图，然后在该图上寻找一棵最小生成树，mst(P) = 6+7 = 13。



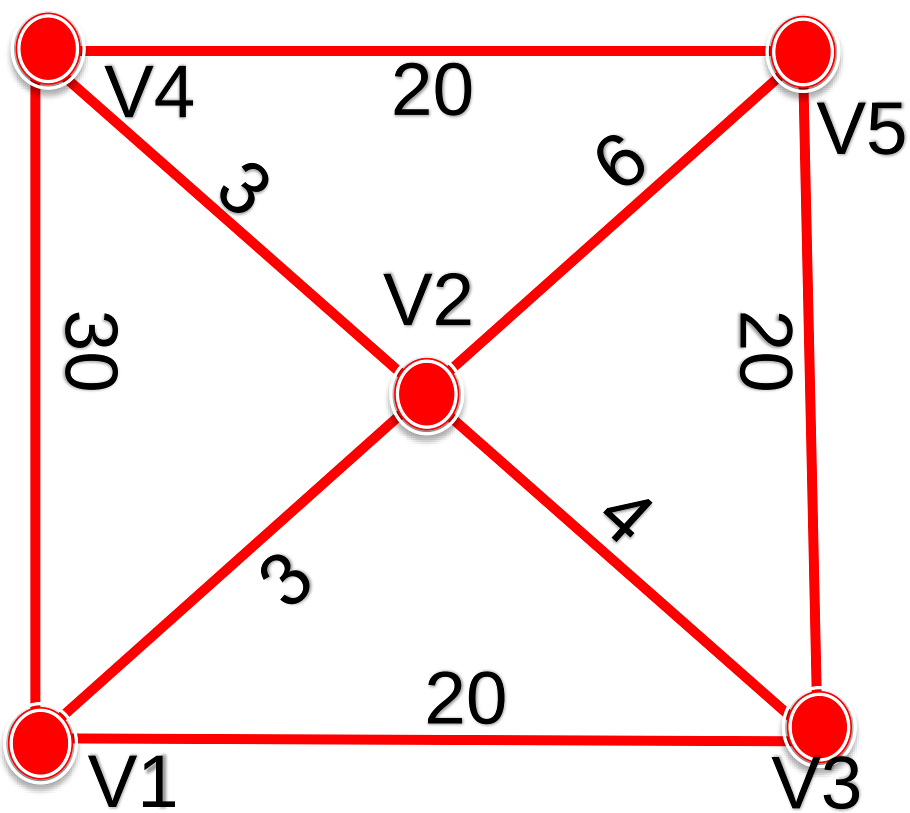
最小生成树算法求解网络斯坦纳树问题

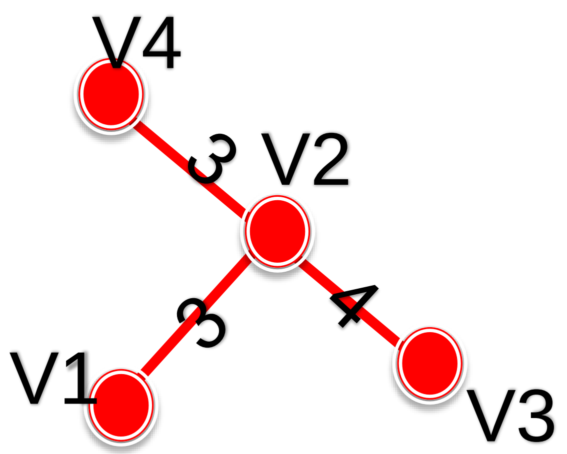
- 3、将其变换到原网络图中，得到最小生成树权重为10。



最小生成树算法求解网络斯坦纳树问题

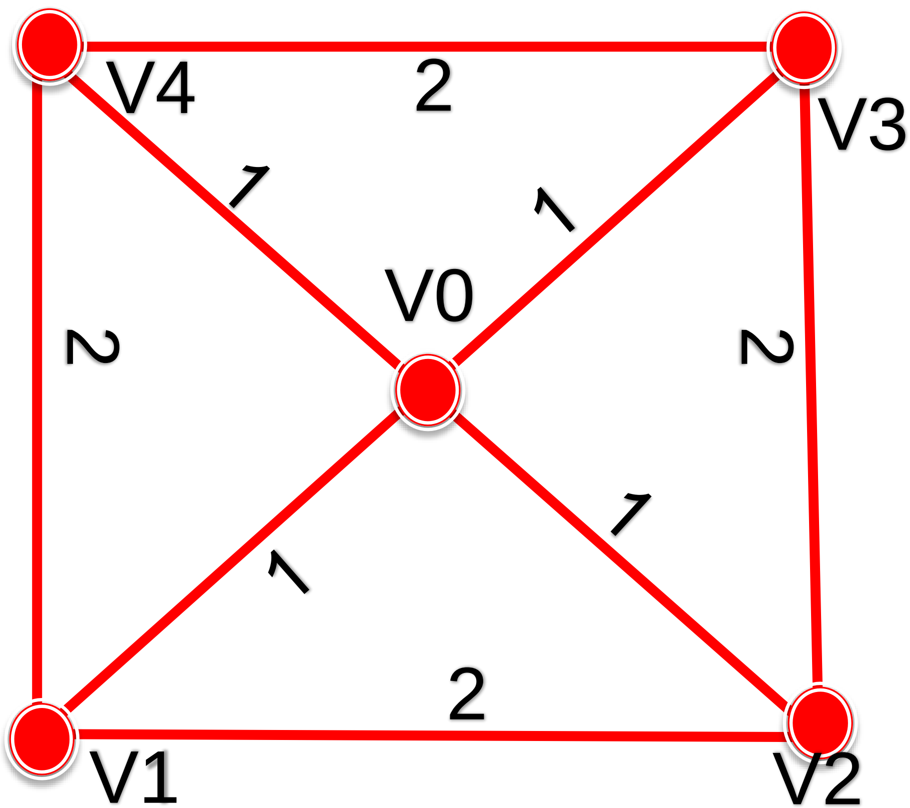
- 在该例子中，采用最小生成树算法求解得到的斯坦纳树恰好是斯坦纳最小树，是否最小生成树能求出最优解呢?





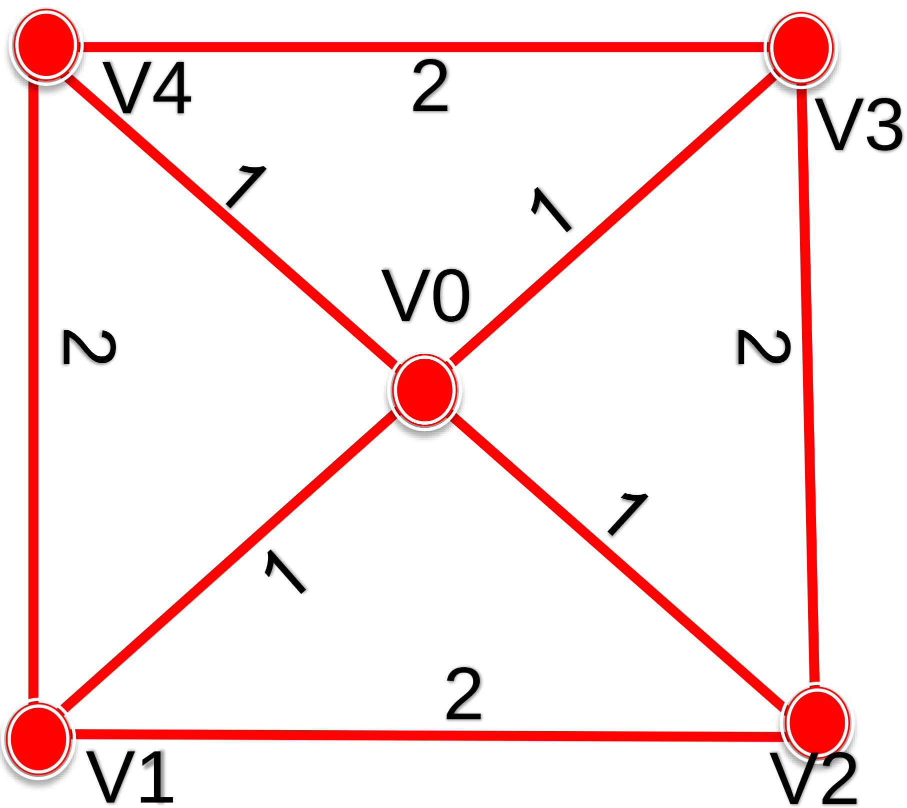
最小生成树算法求解网络斯坦纳树问题

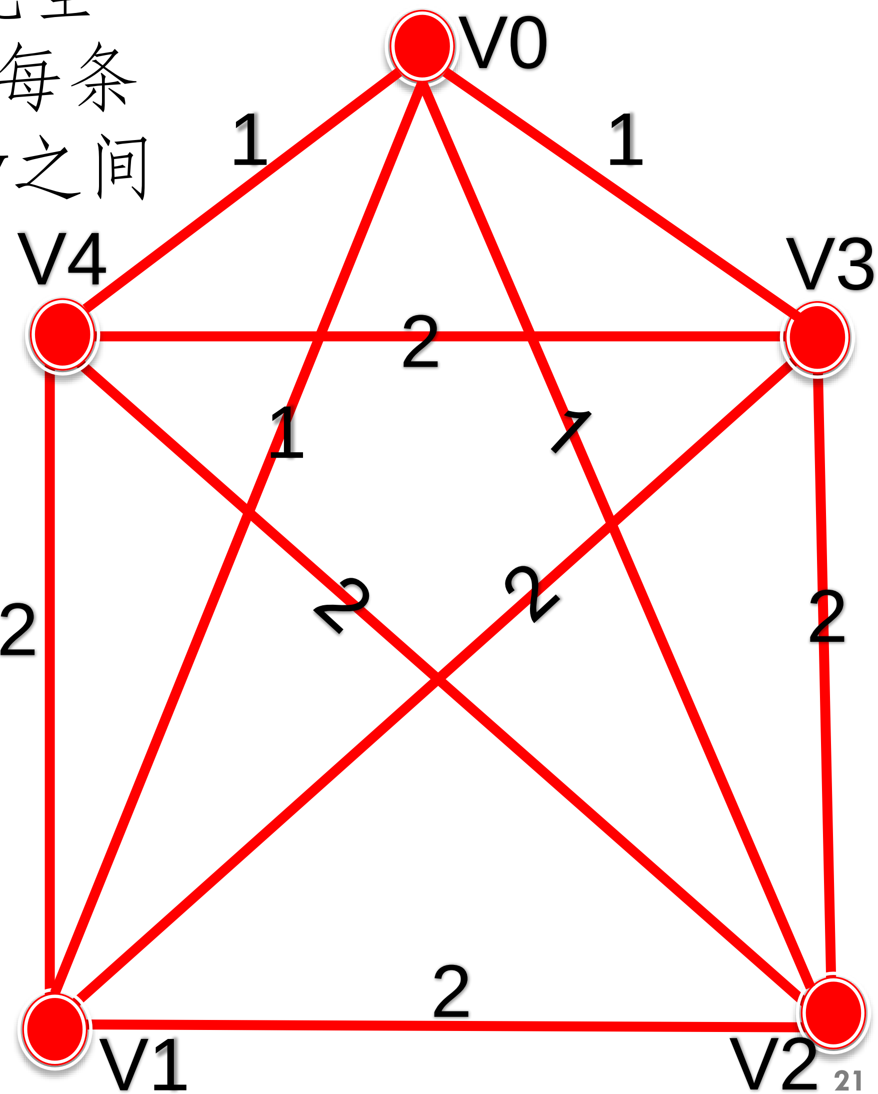
- **例题二：**原始输入图G包含5个点和8条边，求包含结点集合P={V1,V2, V3, V4}的斯坦纳树。



最小生成树算法求解网络斯坦纳树问题

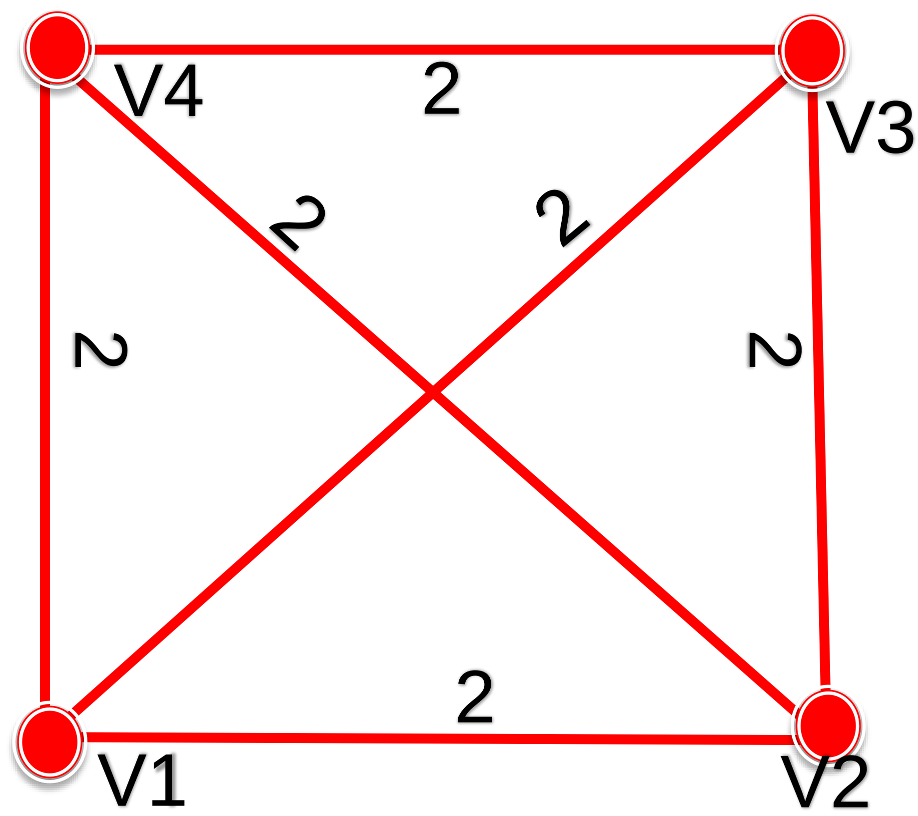
- 1、构造满足三角不等式的完全图：将其转换为完全图，且每条边(u,v)的权定义为原图u和v之间最短路的长度。





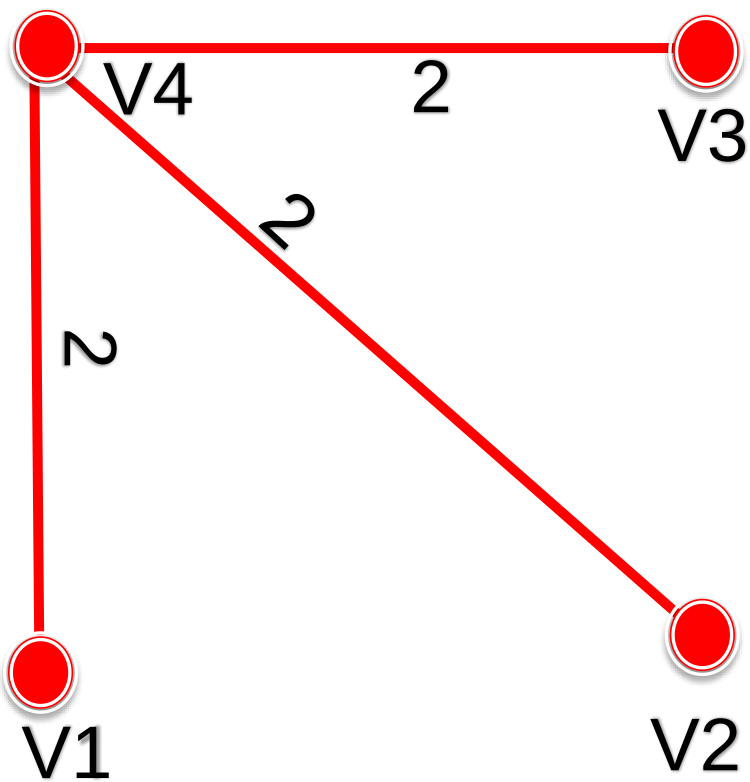
最小生成树算法求解网络斯坦纳树问题

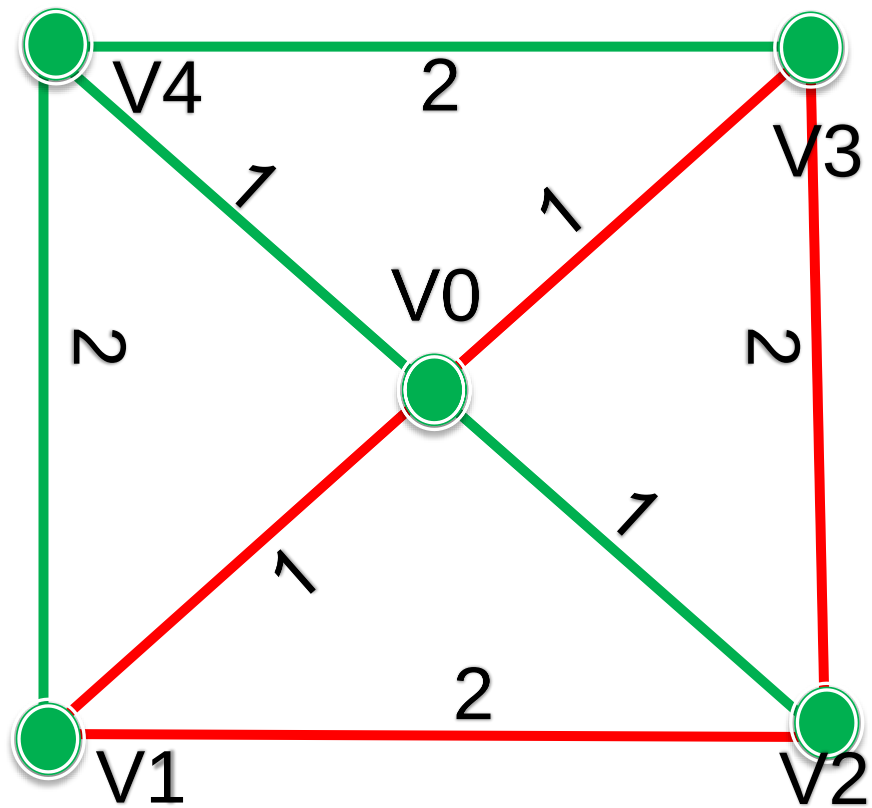
- 2、将完全图转化为一个只包含斯坦纳点P={V1,V2, V3, V4}的完全图，然后在该图上寻找一棵最小生成树，mst(P) = 2+2+2 = 6。



最小生成树算法求解网络斯坦纳树问题

- 3、将其变换到原网络图中，得到最小生成树权重为6。





实际上斯坦纳最小树smt(P) = 4（选则4条权重为1的边），因此，用最小生成树算法求解得到的斯坦纳树不一定是斯坦纳最小树。

最小生成树算法求解网络斯坦纳树问题

- 考虑一个有n+1个点的星图G，每条边的长度都是1。即G是一个完全图，其顶点集合为{v0,v1,…,vn}，边长d(v0,vi) = 1，i = 1,2,…,n；边长 d(vi,vj) = 2，i不等于j，且i,j ∈{1,2,…,n}。在此图中，如果结点集合为P = {v1,v2,…,vn}，那么显然有smt(P) = n, mst(P) = 2(n-1)。当n趋于正无穷大时，则两者比值趋于2。
- 近似算法：现实中许多优化问题无法在多项式时间内找到最优解，因此设计一个算法在多项式时间内找到一个近似最优解（次优解），往往是实际应用中真正有效的手段。

最小生成树算法求解网络斯坦纳树问题

- **近似比**（性能比）：设A是优化问题Π的一个近似算法，算法A在一个输入实例I上的近似比RA(I)被定义为：

$$R_A(I)=\begin{cases}\dfrac{A(I)}{OPT(I)} & if\ \Pi\text{是最小化问题}\\[2ex] \dfrac{OPT(I)}{A(I)} & if\ \Pi\text{是最大化问题}\end{cases}$$

此定义统一使近似比(性能比)RA(I)≥1，越接近1越好

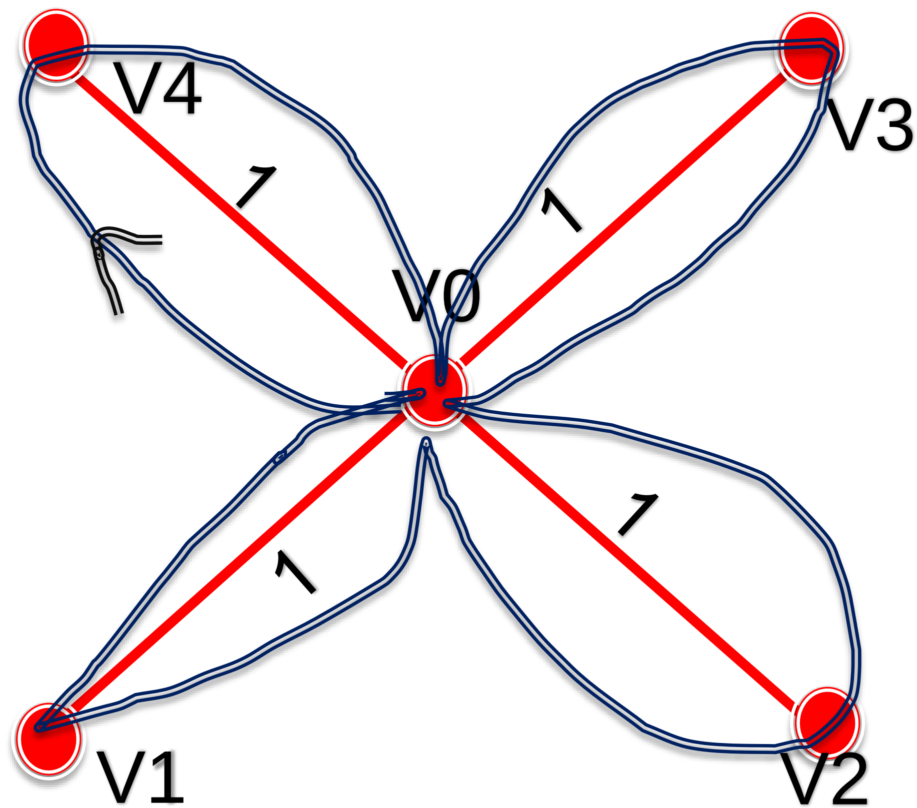
若RA(I) ≤（1+ϵ），则称A是(1+ϵ)—近似算法

最小生成树算法求解网络斯坦纳树问题

- 定理：**对于网络斯坦纳树问题，最小生成树算法的近似比是2。**

举例证明：

- 1、考虑集合P的一棵斯坦纳最小树T，随便选择一个顶点作为起点，对T进行深度优先搜索（DFS），则DFS的顺序恰好经过每条边两次（即存在Euler回路），因此 length(Euler回路) = 2*smt(P);



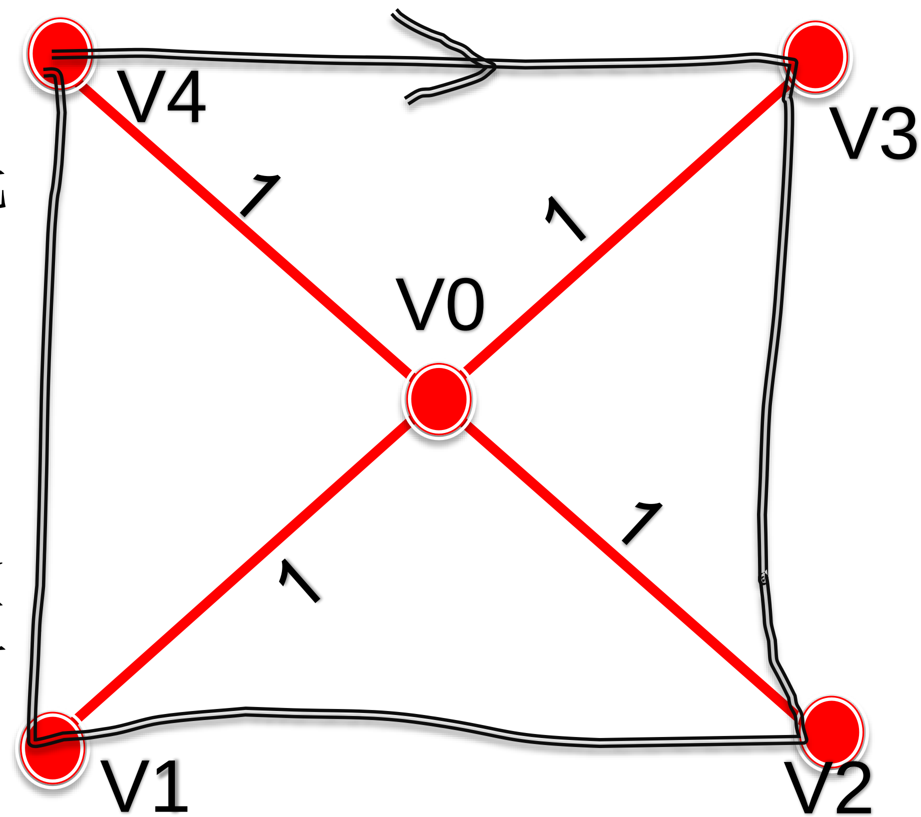
V0-V4-V0-V3-V0-V2-V0-V1-V0

最小生成树算法求解网络斯坦纳树问题

- 定理：对于网络斯坦纳树问题，最小生成树算法的近似比是2。

举例证明：

- 2、按照DFS访问节点的先后顺序，连接结点集P中的节点，就能得到一个结点集的Hamilton圈；显然该Hamilton圈上的每条边与Euler回路上的边有对应关系，由三角不等式知：length(Hamilton圈) ≤ length(Euler回路);



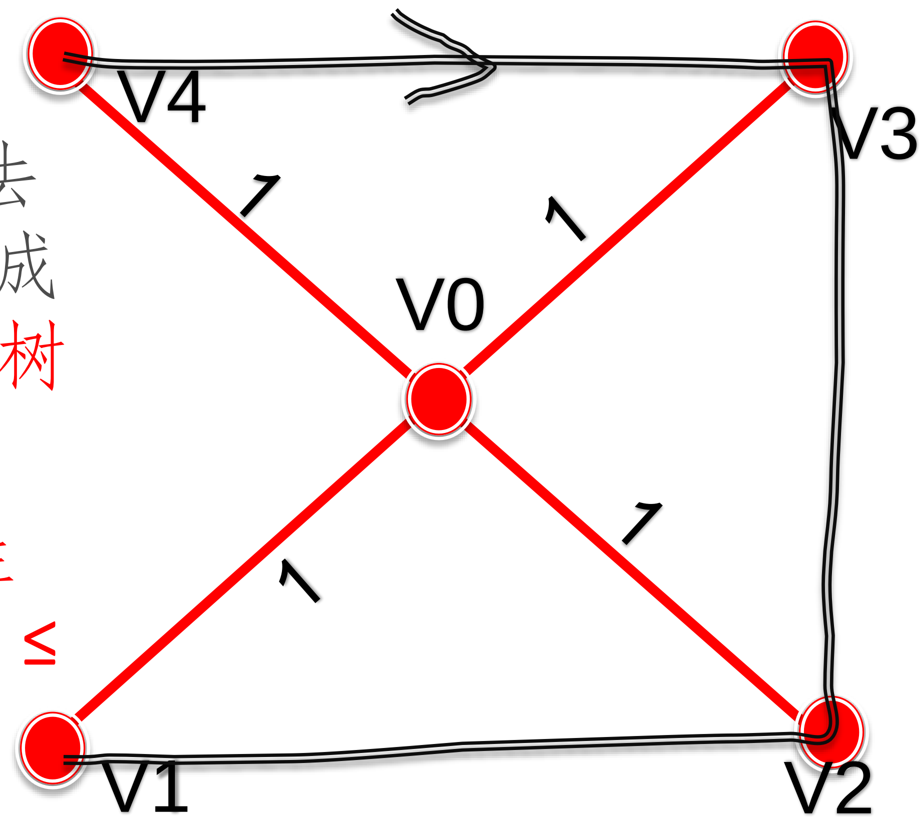
V0-V4-V0-V3-V0-V2-V0-V1-V0

最小生成树算法求解网络斯坦纳树问题

- **定理：对于网络斯坦纳树问题，最小生成树算法的近似比是2。**

举例证明：

- 3、在Hamilton圈上任意去掉一条边就能得到一个生成树，因此有：length(生成树)<length(Hamilton圈)；
- 综上，有mst(P) ≤ length(生成树)<length(Hamilton圈) ≤ length(Euler回路) = 2*smt(P)，得证。



V0-V4-V0-V3-V0-V2-V0-V1-V0

最小生成树算法求解网络斯坦纳树问题

- 定理：对于网络斯坦纳树问题，最小生成树算法的近似比是2。

证明：考虑集合P的一棵斯坦纳最小树T，存在T的一个Euler回路T*，它经过T的每条边恰好两次。沿着该Euler回路T*连接所有结点，则构成一个结点集上的Hamilton圈T′，由于网络斯坦纳树问题满足三角不等式，所以任意一个Hamilton圈的长度都不大于Euler回路；而Hamilton圈去掉一条边就能得到一棵生成树，因此：

mst(P) <length(T′) ≤ length(T*) = 2*smt(P)，得证。

网络斯坦纳树的应用1—多播传输

- 随着Internet的发展与普及，越来越多的场合有一点到多点或者多点到多点的通信需求，比如视频会议[1]、音频会议、远程教育、在线游戏等。相比与多次单播到多播组中的每个用户或者广播该数据包，多播是一个更加高效的传输方式。
- **应用层多播**是在客户端完成多播数据包的复制，每两点之间的数据传输仍然依靠单播进行，这样可以大大降低对路由器的依赖程度，但是却极大的造成了网络拥塞度。
- **IP多播**是一种允许一台或多台主机同时发送数据包到多台主机的网络层技术，通过在路由器上进行数据的复制与多端口转发实现一对多的传输要求，这样就可以使相同的多播数据在链路上只被传输一次，大大降低了网络的负载。

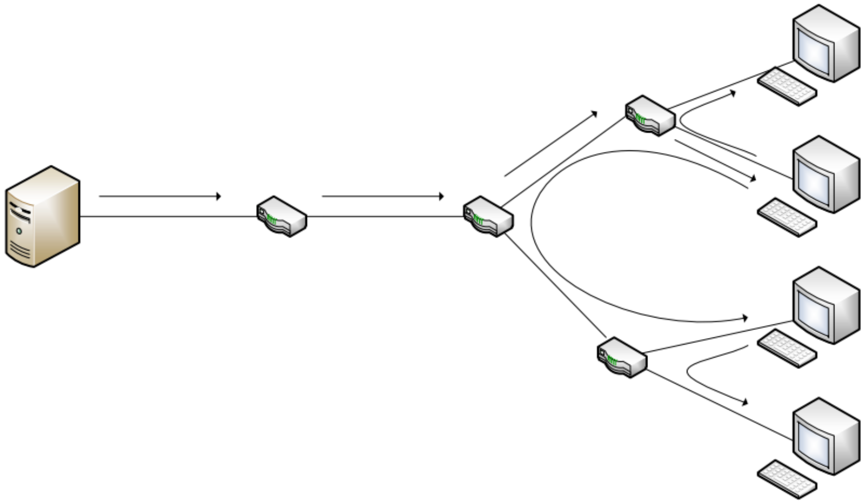


图 2.7 应用层多播

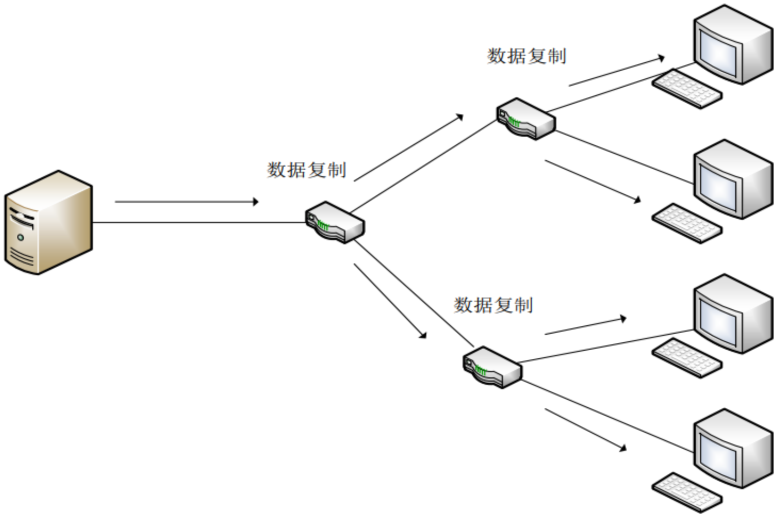


图 2.6 IP 多播

网络斯坦纳树的应用1—多播传输

- **网络模型：** 该网络中包含的主机（host）为H＝{h1，h2，h3，h4，h5…hm}，交换机（switch）为V＝{v1，v2，v3，v4，v5……vn}，其中m为主机的数量，n为交换机的数量，且交换机与主机间的连接情况均可通过控制器进行查询，即链路集合E（edge）也是已知的，因此该网络可以表示成G＝（H U V，E）。在该网络中我们用c(e)和c(v)分别表示链路e最大负载能力（capacity）和交换机v的最大负载能力，其中链路的最大负载能力主要与该链路的带宽有关，交换机的最大负载能力主要与交换机的数据处理能力有关。
- **多播模型：** 在该网络中，将会存在一系列的多播会话（multicast session）$\Gamma=\{\gamma_1, \gamma_2 \ldots\ldots \gamma_q\}$，其中q是多播会话数，每个多播会话可以表示为$\gamma_i=\{s_i, D_i, f_i\}$，其中$s_i$代表源主机（source host），$D_i=\{d_i^1, d_i^2 \ldots\ldots d_i^p\}$代表源主机$s_i$所对应的目的主机（destination hosts）集合，f_i代表该多播会话的带宽需求，比如一个高清视频会议的带宽需求为4Mbps，而一个普通视频会议的带宽需求为2Mbps。

网络斯坦纳树的应用1—多播传输

- 约束：

（1）每个链路的带宽有限

（2）每个交换机的处理能力有限

- 目标：

（1）对于单个多播会话，可以考虑多播树的总代价最小（即斯坦纳最小树问题）；

（2）如果网络中存在多个多播会话呢？

网络斯坦纳树的应用1— 多播传输

□ 存在多个多播会话：

Max-Throughput Multi-Session Multicast: $\max\ \lambda$

$$S.t.\quad \begin{cases} \sum_{\psi_j^i \in \Psi^i} x_{ij} = 1, & \forall \gamma_i \in \Gamma \\ \sum_{\gamma_i \in \Gamma} \sum_{\psi_j^i \ni e} \lambda f_i x_{ij} \le c(e), & \forall e \in E \\ \sum_{\gamma_i \in \Gamma} \sum_{\psi_j^i \ni v} \lambda w_i f_i x_{ij} \le c(v), & \forall v \in V \\ x_{ij} \in \{0,1\}, & \forall \gamma_i \in \Gamma, \forall \psi_j^i \in \Psi^i \end{cases} \qquad (1)$$

□ 首先为每个多播会话r_i利用相关的算法构造若干棵可行多播树，记为多播树集合Ψ^i。x_{ij}表示多播会话r_i是否选择多播树ψ^i_j。第一个方程组表示每一个源主机只能选择一棵多播树；第二个不等式组表示网络拓扑中每条链路传输的数据量不能超过它的最大负载量；第三个不等式组表示网络拓扑中流经每个交换机的数据量不能超过该交换机的最大负载量；第四个条件组表示每一棵多播树要么被选择（此时x_{ij}为1），要么不被选择（此时x_{ij}为0）。而我们的目标是求得当x_{ij}为何值时，λ有最大值。λ表示传输率，比如高清视频的带宽需求为4Mbps，链路可用带宽为2Mbps，则λ为50%。

网络斯坦纳树的应用1— 多播传输

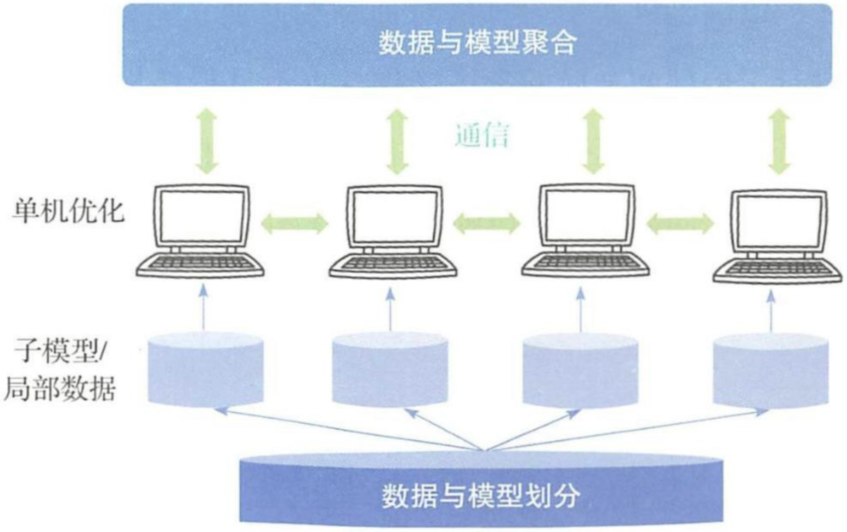
- 如果只存在一个多播会话，但是考虑每个交换机的复制能力有限呢?

答：度受限的斯坦纳最小树问题

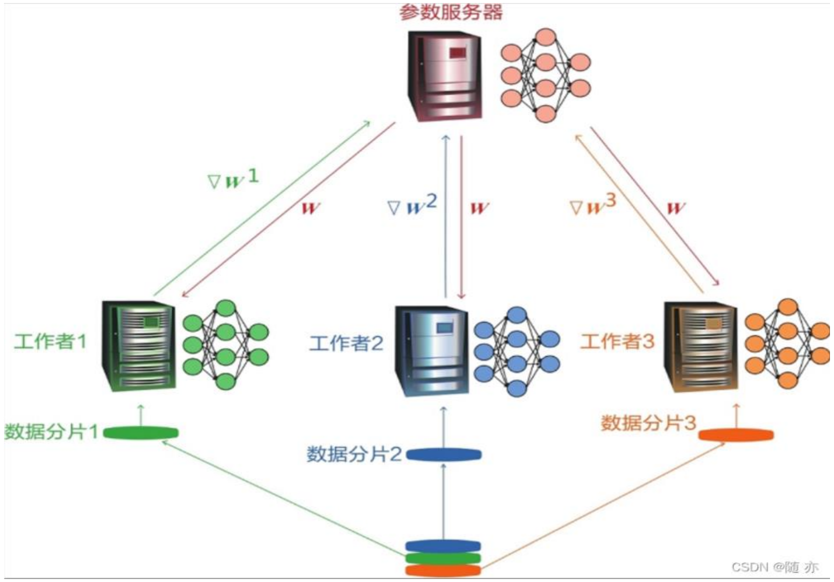
- 如果存在多个多播会话，同时考虑每个交换机的复制能力有限呢?

网络斯坦纳树的应用2—DML任务优化部署

- 机器学习(ML) 是计算机利用已有的数据生成某种模型，并利用此模型预测的一种方法。在确定模型结构之后，根据已知模型寻找模型参数的过程就是训练，训练过程中不断依据训练数据来迭代调整模型的参数值，从而使模型的预测结果更为准确。在现实应用中，要达到好的效果，训练数据集可能很大，模型参数量剧增，会带来很多性能和算法设计问题，单台机器难以胜任，因此分布式机器学习(DML)正在迅速发展。



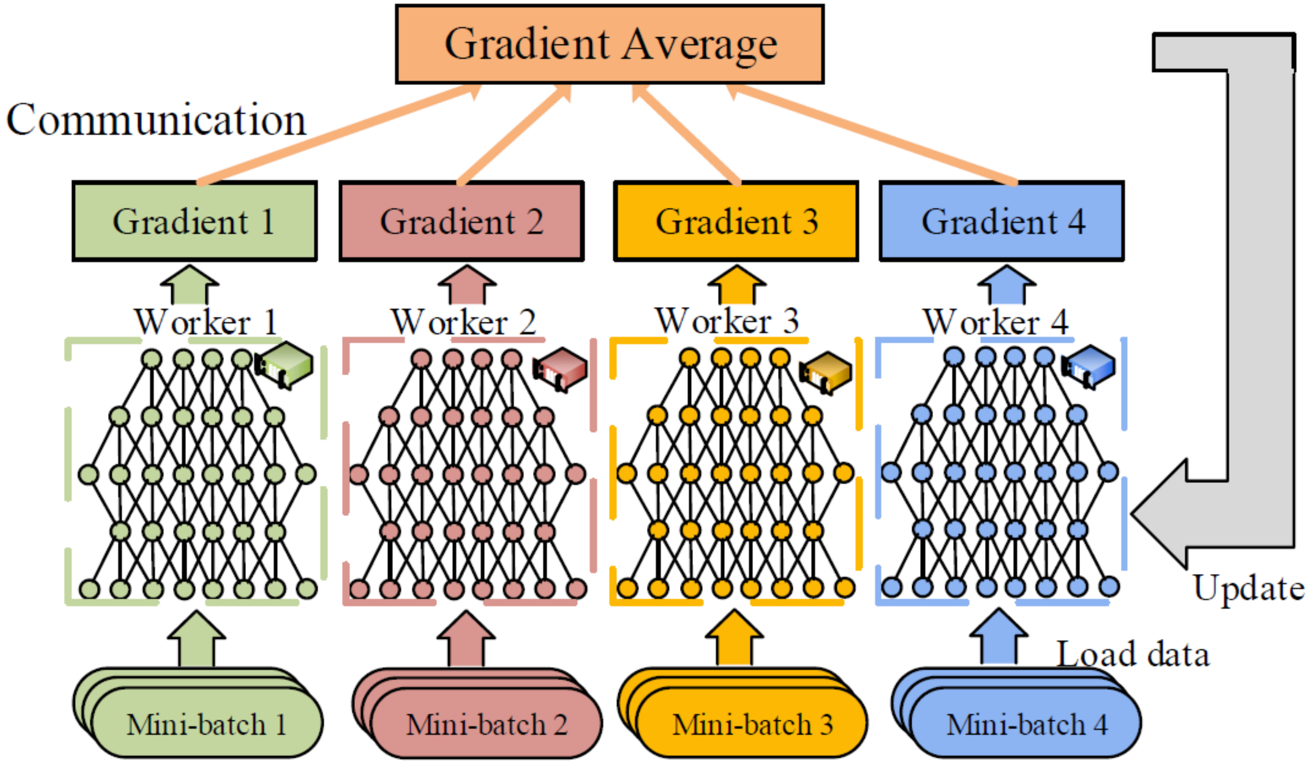
分布式机器学习框架



PS架构下基于数据并行的模型训练

网络斯坦纳树的应用2—DML任务优化部署

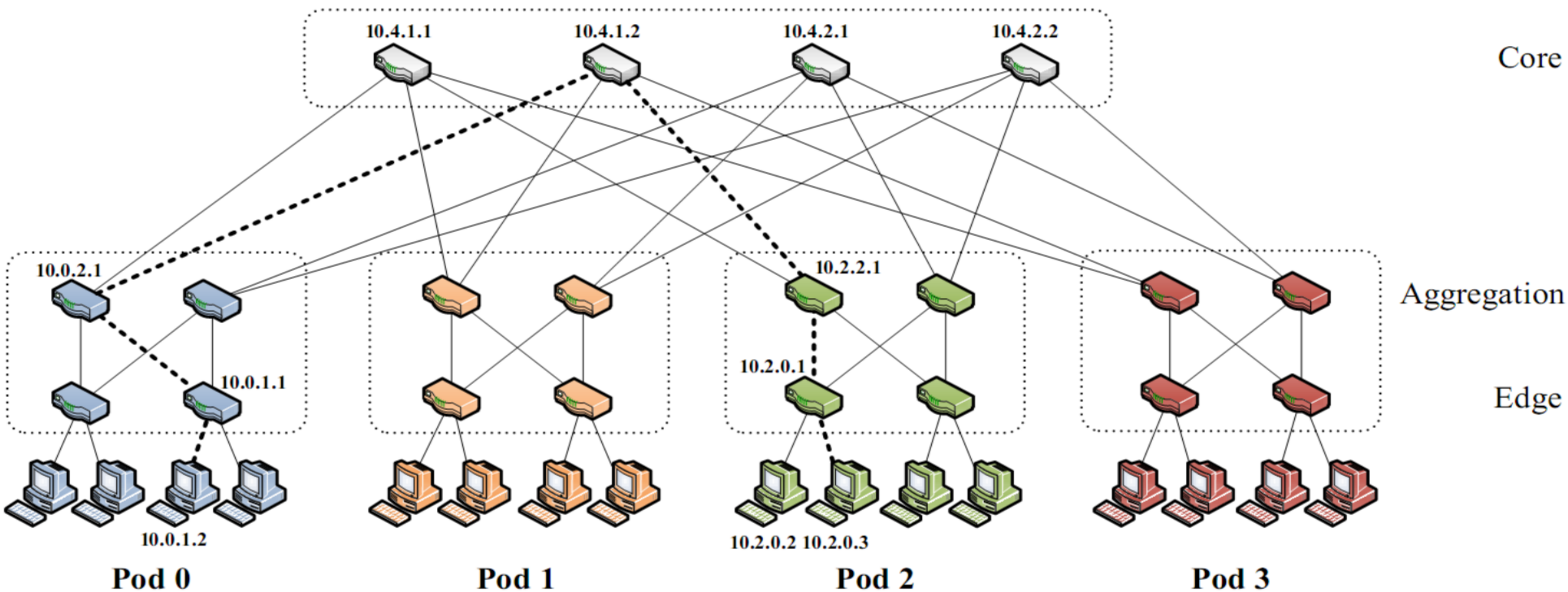
- 数据并行是DML的一种常见的形式，一般分为两个阶段：梯度计算与梯度聚合。每个工作节点(worker)独立训练本地的数据，并发送所得的梯度，在参数服务器(PS)上进行聚合。随后参数服务器将聚合结果返还给工作节点，进行下一轮训练。



PS架构下基于数据并行的模型训练

网络斯坦纳树的应用2—DML任务优化部署

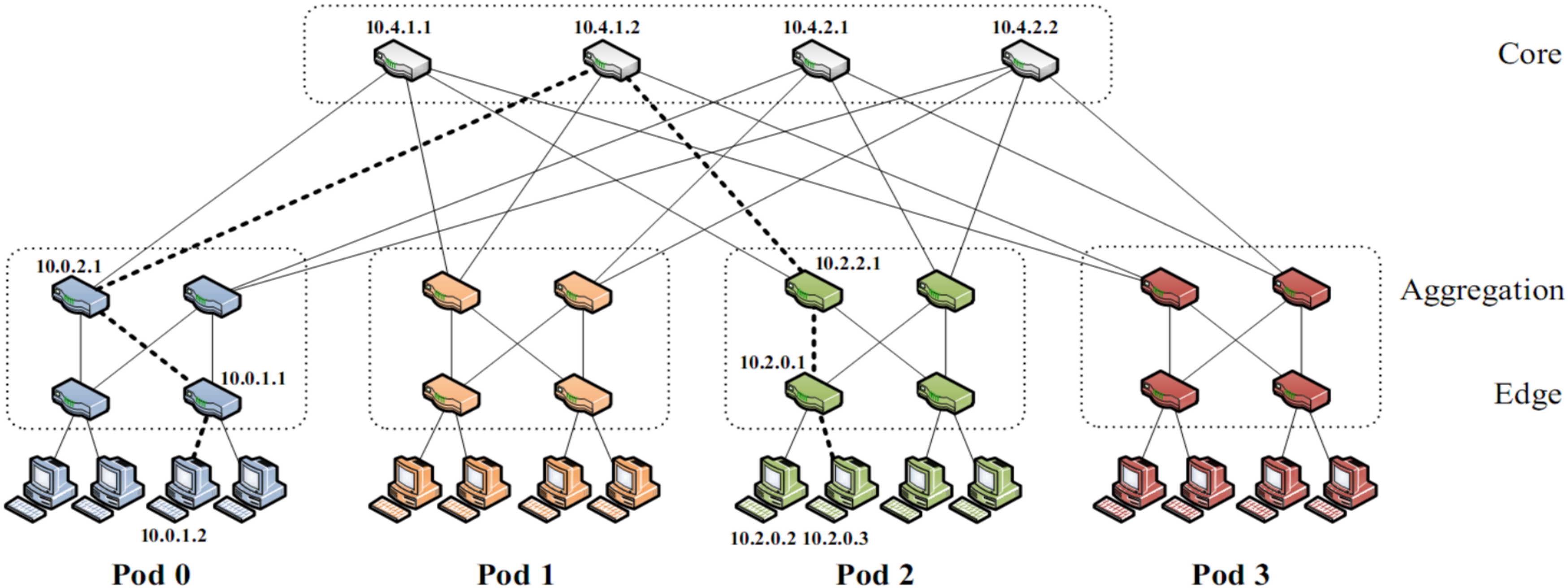
- 考虑到大量的梯度数据传输，梯度聚合过程中的通信时间占用了DML高达50%的总训练时间。因此，降低通信负载对提高模型训练效率至关重要。



典型的Fat-tree数据中心网络拓扑

网络斯坦纳树的应用2—DML任务优化部署

- 1、网络中存在多个可用的服务器，选择哪个服务器作为PS，选择哪些服务器作为worker，进行分布式模型训练呢？
- 2、如何利用可编程交换机（如P4交换机）进行梯度预聚合（in-network aggregation），从而降低通信负载？



典型的Fat-tree数据中心网络拓扑

网络斯坦纳树的应用2—DML任务优化部署

- **网络模型**：服务器集群G包含服务器节点H={h1,h2,…,hm}, 交换机节点V={v1,v2,…,vn}, 其中m为主机的数量，n为交换机的数量。链路集合使用E表示，因此改集群可表示为G = (H U V, E). 所有的服务器规格相同，因此当一个服务器作为PS时，拥有聚合能力B^{PS}。由于其他的任务占用了一部分交换机资源，每一个交换机剩余的处理能力为B_v^{switch}。每一条链路ei每传输一份梯度数据，都有通信量代价c(ei)。一个DML任务需要部署p个worker以及一个PS，并且每个worker以速率r恒定发送。

- **问题定义**：如何选取指定数量的服务器作为PS和worker，并规划梯度聚合的方案，使得部署DML任务产生的系统通信量代价最小。

网络斯坦纳树的应用2—DML任务优化部署

- 约束：

  （1）参数服务器的处理能力有限

  （2）每个交换机的处理能力有限

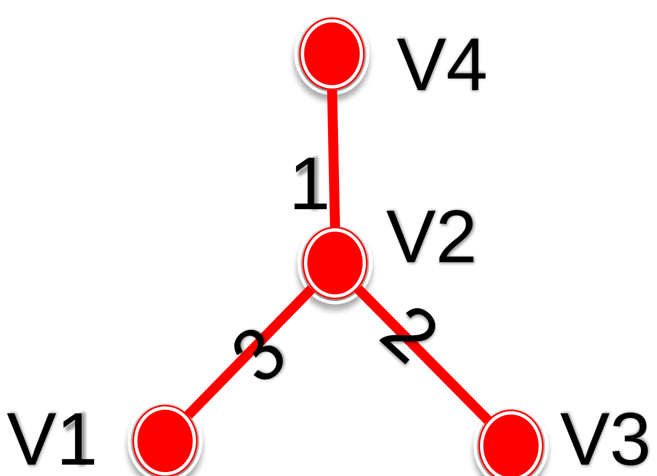
  （3）所选中的worker与PS之间均相互连通

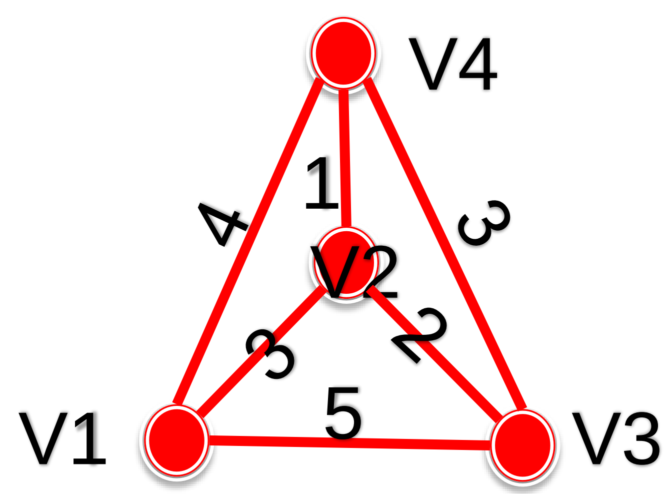
- 目标：

  DML任务产生的系统通信量代价最小

网络斯坦纳树的应用2—DML任务优化部署

- 为了便于描述梯度数据的聚合，我们将网络拓扑进行预处理，将物理的链路转化为逻辑上的聚合路径，表示从一个端点出发的梯度数据，以最短路径逐跳转发至另一个端点，并在这个端点处聚合。
- 我们的目标是选择合适的点进行任务部署，并选择合适的边进行分布式模型训练/聚合，使通信负载（即所选择的树边权）最小。





首先将物理拓扑（例如左图）转换为逻辑上的完全图（例如右图）

网络斯坦纳树的应用2—DML任务优化部署

- 我们用x_v表示是否使用v作为worker/PS/可编程交换机参与聚合，x_e表示是否使用逻辑聚合路径e。
- 优化目标为系统中边上的通信代价之和最小。
- 第一个方程组表示所有选中的节点之间相互连通，这里δ(P)表示满足仅有一个端点在P中的边集合，f(P)表示P是否包含了部分但不是全部的终端节点；
- 第二至四个方程组表示节点的度数限制，即处理能力的限制；
- 第五至七个方程组表示服务器的指派限制，即需要指派一个PS与p个worker，且PS和worker不能重合。

$$\min \sum_{e\in E} x_e \cdot c_e$$

$$\begin{cases}
\sum_{e\in\delta(P)} x_e \geq f(P), & \forall P \subset H \\
\sum_{e\in\delta(\{v\})} x_e \leq \lfloor \frac{B^{PS}}{r} \rfloor \cdot x_v^{PS}, & \forall v \in H \\
\sum_{e\in\delta(\{v\})} x_e = x_v^{worker}, & \forall v \in H \\
\sum_{e\in\delta(\{v\})} x_e \leq \lfloor \frac{B_v^{switch}}{r} \rfloor \cdot x_v^{switch}, & \forall v \in V \\
\sum_{v\in H} x_v^{PS} = 1 & \\
\sum_{v\in H} x_v^{worker} = p & \\
x_v^{PS} + x_v^{worker} \leq 1, & \forall v \in H \\
x_v^{PS} \in \{0,1\}, & \forall v \in H \\
x_v^{worker} \in \{0,1\}, & \forall v \in H \\
x_v^{switch} \in \{0,1\}, & \forall v \in V \\
x_e \in \{0,1\}, & \forall e \in E
\end{cases}$$

网络斯坦纳树的应用2—DML任务优化部署

- 如果考虑worker和PS的位置已知，求如何进行梯度路由，使链路负载最小呢?

（度受限的边权最小的斯坦纳树问题）

- 如果考虑worker和PS的位置已知，网络拓扑已知，如何进行可编程交换机的部署，使部署的交换机数量最小?

（度受限的点权最小的斯坦纳树问题）

网络斯坦纳树的应用2—DML任务优化部署

- 部分经典的斯坦纳树系列问题：
  - STP，最小斯坦纳树问题（边权）；
  - DC-STP，度受限的最小斯坦纳树问题；
  - NW-STP，节点赋权的最小斯坦纳树问题；
  - DC-NW-STP，度受限、节点赋权的最小斯坦纳树问题；
  - PT-STP，partial-terminal最小斯坦纳树问题；
  - NW-PT-STP，节点赋权的partial-terminal最小斯坦纳树问题；

partial-terminal斯坦纳树问题，即在一张图$G=(V,E)$中给定节点集合S，以及集合$S'\subseteq S$。

求出一棵树T，使得其连接S中的所有节点，并且S'中的所有节点必须为树T的叶子节点

STP：Steiner Tree Problem，（最小）斯坦纳树问题；
DC：Degree Constrained，度受限；
NW：Node Weighted，节点赋权；
PT：Partial Terminal；