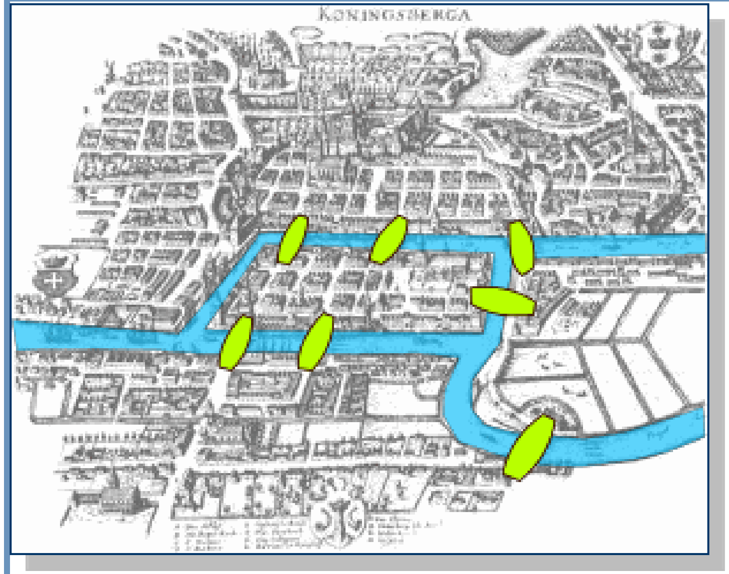
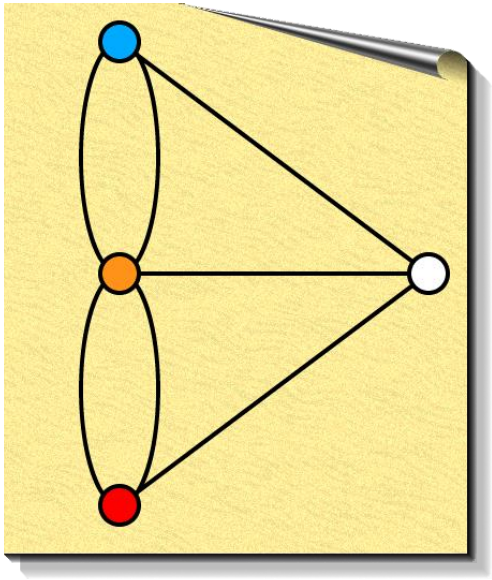



# 一种针对未知需求的域内路由算法

## 理解网络 LP 建模与对偶方法在网络问题的应用

汇报人：黄卓彬
指导老师：王雄副教授

**October 13  2021**



# Background

## Reference Work:

*Tabatabaee, V., Kashyap, A., Bhattacharjee, B., La, R.J. and Shayman, M., 2006. Robust routing with unknown traffic matrices.*

## Motivation & Background 分析:

- 在 **Datacenter** 中，端到端通信带宽需求通常并不是固定的，而是动态可变的
- 尚无可编程控制平面，网络控制面凭借分布式控制协议收敛

*Casado, Martin, et al. "Ethane: Taking control of the enterprise." ACM SIGCOMM computer communication review 37.4 (2007): 1-12.*

- 尚无可编程数据平面，数据面实时性能测量尚不可行

*Bosshart, Pat, et al. "P4: Programming protocol-independent packet processors." ACM SIGCOMM Computer Communication Review 44.3 (2014): 87-95.*

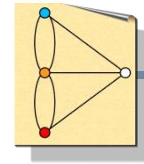
- 必须基于可变的流量需求尝试离线建立多商品流的路由优化策略

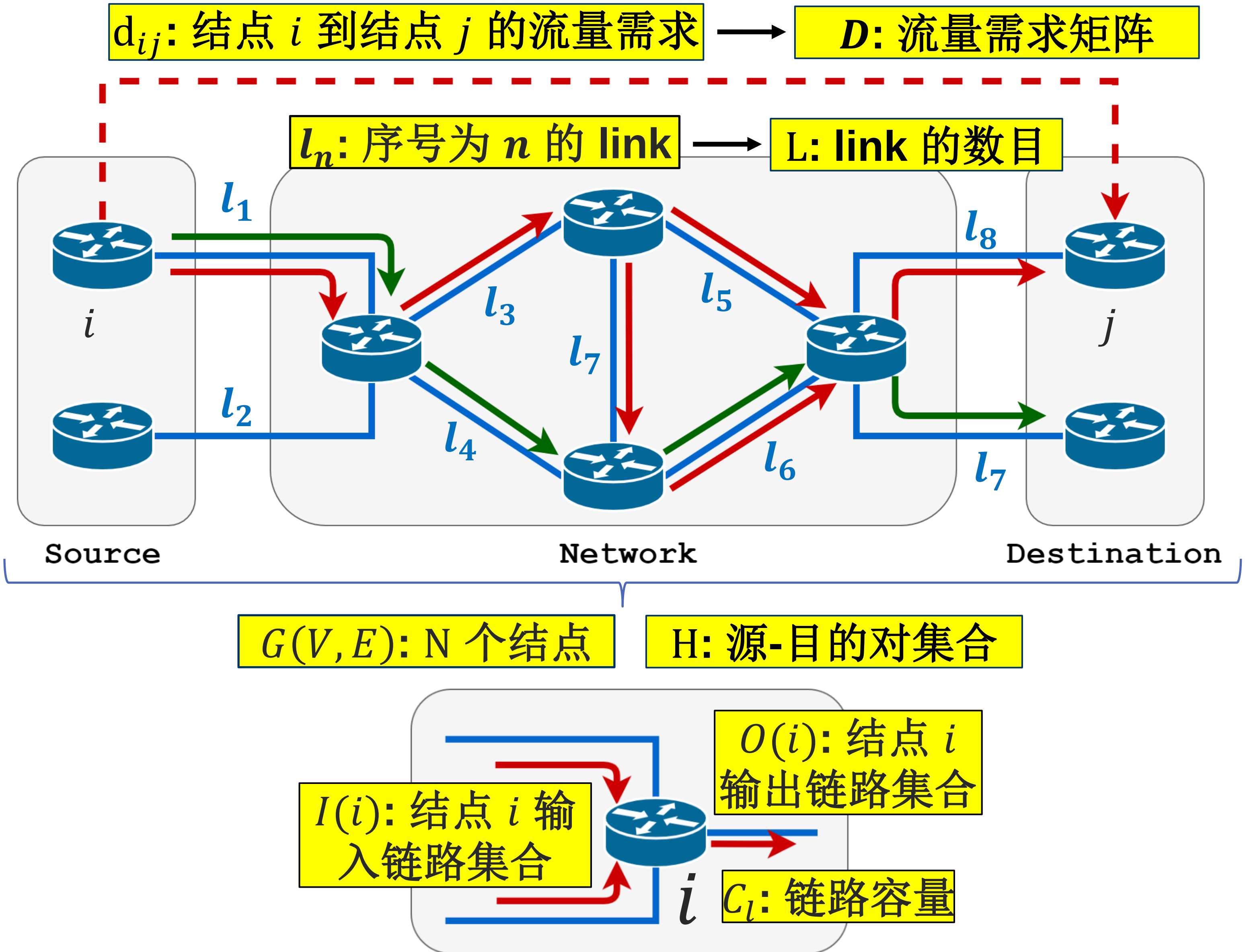
# Content

我们将会介绍：

- **Background**：动态变化的需求在 Datacenter 网络中是常见的场景
- **Node-Link Modeling**：如何对一个需求变化的流量矩阵进行建模
- **Algorithm**：如何求解一个 Linear Semi-infinite Programming 问题
- **Dual**：如何运用对偶思想来简化我们的算法流程
- **Link-Path Modeling**：将我们的模型转化为 Path-based 的建模方法

# Notation





$d_{ij}$: 结点 $i$ 到结点 $j$ 的流量需求
$D$: 流量需求矩阵
$l_n$: 序号为 $n$ 的 link
L: link 的数目
$l_1$
$l_2$
$l_3$
$l_4$
$l_5$
$l_6$
$l_7$
$l_8$
$l_7$
$i$
$j$
Source
Network
Destination
$G(V,E)$: N 个结点
H: 源-目的对集合
$O(i)$: 结点 $i$ 输出链路集合
$I(i)$: 结点 $i$ 输入链路集合
$i$
$C_l$: 链路容量

# Node-Link Constrains

$f_{ij}(l)$: 需求对 $(i,j)$ 在 $l$ 上摊销的流量比例

## 流守恒约束

$$\sum_{l\in O(k)} f_{ij}(l) - \sum_{l\in I(k)} f_{ij}(l) = 0 \quad k \neq i,j$$

$$\sum_{l\in O(k)} f_{ij}(l) - \sum_{l\in I(k)} f_{ij}(l) = 1 \quad k = i$$

$c(l)$: 链路$l_n$ 的容量

链路利用率 $\boxed{u(l) = \sum_{ij} \frac{f_{ij}(l)\cdot d_{ij}}{c(l)}}$

## 链路容量约束

$$u(l) = \sum_{ij} \frac{f_{ij}(l)\cdot d_{ij}}{c(l)} \leq 1 \qquad l \in L \,\&\, d \in D$$

# Node-Link Modeling

## LP Modeling [min-max problem]

Minimize t **min-max 链路利用率目标**

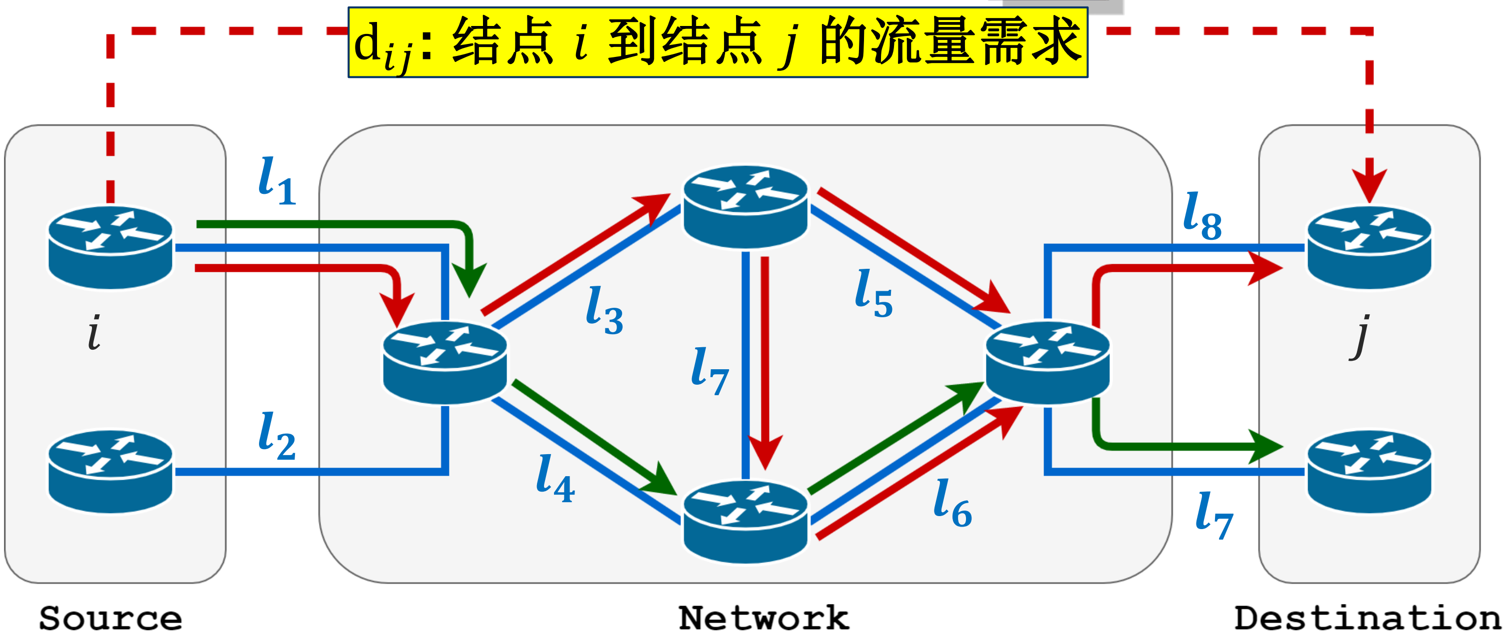
subject to

$f_{ij}(l)$ is a routing, $\forall (i,j) \in H$ **流守恒约束**

$\sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)} \le t, l \in L \;\&\; d \in D$ **链路容量约束**

- 决策变量和系数矩阵分别是？ ➡ $f_{ij}(l)$ 和 $d_{ij}$
- 有没有什么不太对的地方？ ➡ $\boldsymbol{d_{ij}}$ 是非固定的
- 怎么表达非固定流量需求 $\boldsymbol{d_{ij}}$ ？ ➡ 给定一个范围

# Node-Link Modeling




软管道 (Hose) 模型

| | |
|---|---|
| 表达 | $\sum_{k=1}^{N} d_{ik} \leq \eta_i, \mathrm{i} = 1, \dots, \mathrm{N}$<br>$\sum_{k=1}^{N} d_{kj} \leq \lambda_\mathrm{j}, j = 1, \dots, N$ |
| 描述 | 设置需求源 $i$ 的流出流量上限 $\eta_i$<br>设置需求目的 j 的流入流量上限 $\lambda_\mathrm{j}$ |
| 工程解释 | 网络设备物理特性本身的限制：网络节点出口、入口带宽 (物理网络限制) |

硬管道 (Pipe) 模型

| | |
|---|---|
| 表达 | $d_{ij} \leq w_{ij} \quad i, j = 1, \dots, N$ |
| 描述 | 设置需求对 $(i, j)$ 的流量上限 $\mathrm{w}_{ij}$ |
| 工程解释 | 网络管理员会事先设置流量路由规则，使用 SLA 等来限制流量规模 (虚拟网络限制) |

# Node-Link Modeling

## 我们现在得到了什么？

### LP 问题模型

**Minimize** $t$

**subject to**

$f_{ij}(l)$ is a routing, $\forall (i,j) \in H$

$$\sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)} \leq t, l \in L \,\&\, d \in D$$

### 系数矩阵范围

*Hose Model:*

$$\sum_{k=1}^{N} d_{ik} \leq \eta_i, \mathrm{i} = 1, \dots, \mathrm{N}$$
$$\sum_{k=1}^{N} d_{kj} \leq \lambda_j, j = 1, \dots, N$$

*Pipe Model:*

$$d_{ij} \leq w_{ij} \quad i,j = 1, \dots, N$$

## 具有一定动态范围的 $\mathrm{d}_{ij}$导致了什么？

- 无限的约束条件 (Linear Semi-Infinite Programing)

## 怎样将问题转化为我们能求解的形式？

- Exchange Method

# Algorithm

从 $D$ 中选取一个非零需求矩阵 $D_0$

$$\text{Minimize } t$$

**subject to**

$$f_{ij}(l) \text{ is a routing}, \forall(i,j) \in H$$

$$\sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)} \le t, l \in L \;\&\; d \in D_0$$

我们得到了什么？

- 求解得到一个基于 $D_0$ 的 min-max 链路利用率 $u^*$
- 得到 $u^*$和 $D_0$ 对应的端到端路由分配策略 $f_{ij}{}^*(l)$

思考：

1. 我们得到的 $f_{ij}{}^*(l)$ 能否满足所有的通信需求呢？
2. 我们得到的 $u^*$是否是最优的 min-max 链路利用率呢？

$$\text{Maximize } \sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)}$$

**subject to**

$$\sum_{k=1}^{N} d_{ik} \le \eta_i, i = 1, \dots, N$$

$$\sum_{k=1}^{N} d_{kj} \le \lambda_j, j = 1, \dots, N$$

$$d_{ij} \le w_{ij} \quad i,j = 1, \dots, N$$

我们得到了什么？

- 求解得到一个基于 $f_{ij}{}^*(l)$ 的 max 链路利用率 $u'^*$
- 导致了 $u'^*$ 的某个需求矩阵 $D'_0$
- 产生 $u'^*$ 负载的某条链路 $l'$

思考：

1. 如果 $u'^* \le u^*$，说明什么问题？
2. 如果 $u'^* > u^*$，说明什么问题？

如果 $u'^* > u^*$，则向原问题中加入约束：

$$\sum_{ij} \frac{f_{ij}(l') \cdot d_{ij}}{c(l')} \le t, d \in D'_0$$

10/27/2021

# Why Dual

## 问题解决了，但是…

**First LP**

Minimize t

subject to

$f_{ij}(l)$ is a routing, $\forall (i,j) \in H$

$\sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)} \le t, l \in L \; \& \; d \in \boldsymbol{D_0}$

**Second LP**

Maximize $\sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)}$

subject to

$\sum_{k=1}^{N} d_{ik} \le \eta_i \,, \mathrm{i} = 1, \ldots, \mathrm{N}$

$\sum_{k=1}^{N} d_{kj} \le \lambda_j \,, j = 1, \ldots, N$

$d_{ij} \le \; w_{ij} \quad i,j = 1, \ldots, N$

- 在我们讨论算法中，包含了两个相互依赖的 LP 问题
- 如果能够合并成为一个 LP 问题，算法将更高效！

## 能够直接把两个问题合并起来吗？

- 思考我们之前为什么要拆分成为两个问题？
- 直接合并将把我们带回 Linear Semi-Infinite Programing！

## 如何合并？

对偶的 Motivation

- 把 Second LP 转化成其对偶形式，消除 First-LP 和 Second-LP 的关联性

# How Dual

## A story as appetizer…

| | Resource 1 | Resource 2 | Resource 3 | 产品价格 |
|---|---|---|---|---|
| Product 1 | 10 units | 20 units | 10 units | 60 dollars |
| Product 2 | 30 units | 10 units | 5 units | 70 dollars |
| 资源总量 | 120 units | 100 units | 80 units | |

### 原问题

**Producer：** 生产 **Products**，最大化收益

**Maximize** $60x_1 + 70x_2$

**subject to**

$$10x_1 + 30x_2 \le 120$$
$$20x_1 + 10x_2 \le 100$$
$$10x_1 + 5x_2 \le 80$$
$$x_1, x_2 \ge 0$$

### 对偶问题

**Consumer：** 购买 **Resources**，最小化支出

***Minimize*** $120y_1 + 100y_2 + 80y_3$

**subject to**

$$10y_1 + 20y_2 + 10y_3 \ge 60$$
$$30y_1 + 10y_2 + 5y_3 \ge 70$$
$$y_1, y_2, y_3 \ge 0$$

# How Dual

## Observation

- 对于 Maximize 的 LP，问题的构建者是利用 Resource 生产 Product 的企业 A，它要最大化 Product 产生的收益，但是无奈 Resource 有限，它需要确定各种 Product 生产的数量
- 对于 Minimize 的 LP，问题的构建者是想从企业 A 直接购买 Resource 的企业 B，它要最小化购买 Resource 产生的支出，但是无奈它不能让企业 A 亏本，它需要确定为各种 Resource 的定价

## Inspiration：回到我们的 Second LP

**Second LP**

$$\text{Maximize} \sum_{ij} \frac{f_{ij}(l)\cdot d_{ij}}{c(l)}$$

subject to

$$\sum_{k=1}^{N} d_{ik} \le \eta_i, \ i = 1, \dots, N$$
$$\sum_{k=1}^{N} d_{kj} \le \lambda_j, \ j = 1, \dots, N$$
$$d_{ij} \le w_{ij} \quad i, j = 1, \dots, N$$

问题的构建者是谁？

- 端到端应用，它生产了各种网络流量 (Product)

它要最大化什么？

定义 $p_{ij}(l) = \frac{f_{ij}(l)}{c(l)}$：单位 $d_{ij}$ 流在链路 $l$ 上产生的收益

- 所有网络流在链路 $l$ 上的总收益 $\sum_{ij} p_{ij}(l) \cdot d_{ij}$

它受限于？ ➡ 物理出口带宽 $\eta_i$ 、物理入口带宽 $\lambda_j$ 、端到端链路带宽 $w_{ij}$ (Resource)

它要确定？ ➡ 各条流的大小 $d_{ij}$ (数量)

# How Dual

## Inspiration：对于 Second LP 的对偶问题呢？

问题的构建者是谁？

- 收购 网络基础设施 (Resource) 的人

它要最小化什么？ 物理出口、物理入口 & 端到端链路

- 为链路 $l$ 收购网络基础设施的总支出

它受限于？

- 对于各个需求对，在链路 $l$ 上的支出不能低于端到端应用收益 (收入)

它要确定？

- 单位基础设施的定价 $r_i(l), \lambda_j(l), q_{ij}(l)$

Second LP (Origin)

Maximize $\sum_{ij} \frac{f_{ij}(l)\cdot d_{ij}}{c(l)}$

**subject to**

$$\sum_{k=1}^{N} d_{ik} \le \eta_i, i = 1, \dots, N$$
$$\sum_{k=1}^{N} d_{kj} \le \lambda_j, j = 1, \dots, N$$
$$d_{ij} \le w_{ij} \quad i,j = 1, \dots, N$$

Second LP (Dual)

Minimize $\sum_i \eta_i r_i(l) + \sum_j \lambda_j y_j(l) + \sum_{ij} w_{ij} q_{ij}(l)$

**subject to**

$$r_i(l) + y_j(l) + w_{ij}(l) \ge \frac{f_{ij}(l)}{c(l)}$$
$$r_i(l), y_j(l), q_{ij}(l) \ge 0$$

# Algorithm

## 我们现在得到了什么？

**First LP**

**Minimize** t

**subject to**

$f_{ij}(l)$ is a routing, $\forall (i,j) \in H$

$\sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)} \le t, l \in L \ \& \ d \in D_0$

**Maximize** $\sum_{ij} \frac{f_{ij}(l) \cdot d_{ij}}{c(l)}$

**subject to**

$\sum_{k=1}^{N} d_{ik} \le \eta_i, \mathrm{i} = 1, \dots, \mathrm{N}$

$\sum_{k=1}^{N} d_{kj} \le \lambda_j, j = 1, \dots, N$

$d_{ij} \le w_{ij} \quad i,j = 1, \dots, N$

**Second LP (Origin)**

**Minimize** $\sum_i \eta_i r_i(l) + \sum_j \lambda_j y_j(l) + \sum_{ij} w_{ij} q_{ij}(l)$

***subject to***

$r_i(l) + r_i(l) + r_i(l) \ge \frac{f_{ij}(l)}{c(l)}$

$r_i(l), y_j(l), q_{ij}(l) \ge 0$

**Second LP (Dual)**

**Minimize** $t$

**subject to**

$f_{ij}(l)$ is a routing, $\forall i,j,l$

$$\sum_i \eta_i r_i(l) + \sum_j \lambda_j y_j(l) + \sum_{ij} w_{ij} q_{ij}(l) \le t, \forall l$$

$$f_{ij}(l) - c(l)[r_i(l) + r_i(l) + r_i(l)] \ge 0, \forall l,i,j$$

$$r_i(l), y_j(l), q_{ij}(l) \ge 0, \forall l,i,j$$

**Final LP**

第二个约束如何得到的？

➡ 弱对偶定理

# Why Link-Path

## Node-Link Modeling 有什么问题呢？

**Final LP**

Minimize $t$

subject to

$$f_{ij}(l) \text{ is a routing}, \forall i,j,l$$

$$\sum_i \eta_i r_i(l) + \sum_j \lambda_j y_j(l) + \sum_{ij} w_{ij} q_{ij}(l) \le t, \forall l$$

$$f_{ij}(l) - c(l)[r_i(l) + r_i(l) + r_i(l)] \ge 0, \forall l,i,j$$

$$r_i(l), y_j(l), q_{ij}(l) \ge 0, \forall l,i,j$$

- 对于一个现实的网络问题来说，我们更偏向于基于 path 来思考和配置网络策略：
  - 一对需求对之间的 path 数
  - 每条 path 的跳数限制
  - 令每条 path 都经过某一个测量节点
  - ……

- 约束过多，考虑一个全连接的网络【# 源目的对: $N(N-1)$】：

  \# $f_{ij}(l)$ is a routing, $\forall (i,j) \in H$ ： $N(N+M)(N-1)$

  \# $\sum_i \eta_i r_i(l) + \sum_j \lambda_j y_j(l) + \sum_{ij} w_{ij} q_{ij}(l) \le t$: $M$

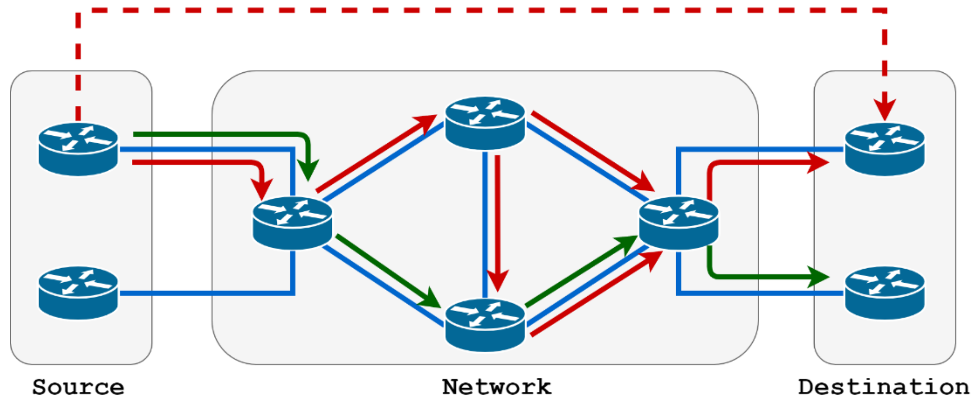
  \# $f_{ij}(l) - c(l)[r_i(l) + r_i(l) + r_i(l)] \ge 0$: $MN(N-1)$

  \# $r_i(l), y_j(l), q_{ij}(l) \ge 0$: $MN(N-1) + 2NM$

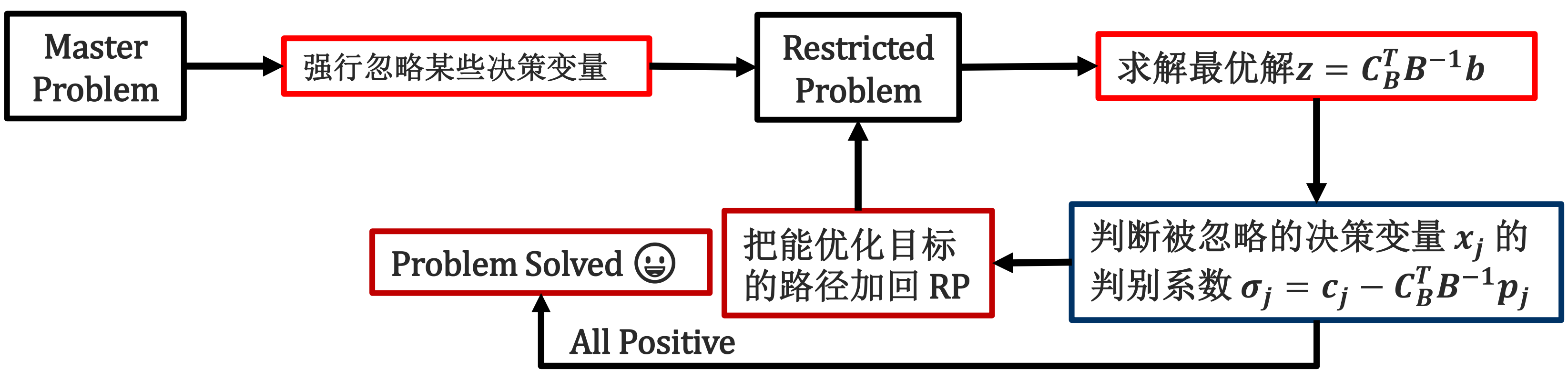
# How Link-Path

## 能直接基于 Path 来进行建模吗？

- 随着网络规模 $G(V,E)$ 的增大，Path 的数量将会指数爆炸式地增长
- 而 Path 的数量影响的是决策变量的个数
- 事实是，通常在最优解中，仅有一小部分的 path 会承担流量负载
- 仅有一小部分的决策变量会对优化目标做出贡献




## Column generation (列生成) 算法的思路：



# Link-Path Constraints

- $x_p$: 需求对 $(i,j)$ 在路径 $p$ 上摊销的流量比例

  $P^k$: 在第 k 次迭代时的活跃路径集 (active path set)

  $P_{ij}^k$: 在第 k 次迭代时从 $i$ 到 $j$ 的活跃路径集 (active path set)

**需求量约束**

$$\sum_{p \in P_{ij}^k} x_p = 1, \forall (i,j) \in H$$

- $\sqcap_l^k$: 在第 k 次迭代时的经过链路 $l$ 的 $P^k$ 的子集

  $f_{ij}^k(l)$: 在第 k 次迭代时需求对 $(i,j)$ 在链路 $l$ 上摊销的流量比例

**“链路容量约束”**

$$f_{ij}^k(l) = \sum_{p \in (P_{ij}^k \cap \sqcap_l^k)} x_p, \forall (i,j) \in H, l \in L$$

10/27/2021

# Link-Path Modeling

## 将 Final-LP 进行进行替换：

$$
\begin{aligned}
&\text{Minimize } t \\
&\textbf{subject to} \\
&f_{ij}(l) \text{ is a routing}, \forall i,j,l \\
&\sum_i \eta_i r_i(l) + \sum_j \lambda_j y_j(l) + \sum_{ij} w_{ij} q_{ij}(l) \le t, \forall l \\
&f_{ij}(l) - c(l)[r_i(l) + r_i(l) + r_i(l)] \ge 0, \forall l,i,j \\
&r_i(l), y_j(l), q_{ij}(l) \ge 0, \forall l,i,j
\end{aligned}
$$

Node-Link Modeling

$$
\begin{aligned}
&\text{Minimize } t \\
&\textbf{subject to} \\
&\sum_{p \in P_{ij}^k} x_p = 1, \forall (i,j) \in H \\
&\sum_i \eta_i r_i(l) + \sum_j \lambda_j y_j(l) + \sum_{ij} w_{ij} q_{ij}(l) \le t, \forall l \\
&\sum_{p \in (P_{ij}^k \cap \Pi_l^k)} x_p - c(l)[r_i(l) + r_i(l) + r_i(l)] \ge 0, \forall l,i,j \\
&r_i(l), y_j(l), q_{ij}(l), x_p \ge 0, \forall l,i,j,p
\end{aligned}
$$

Link-Path Modeling

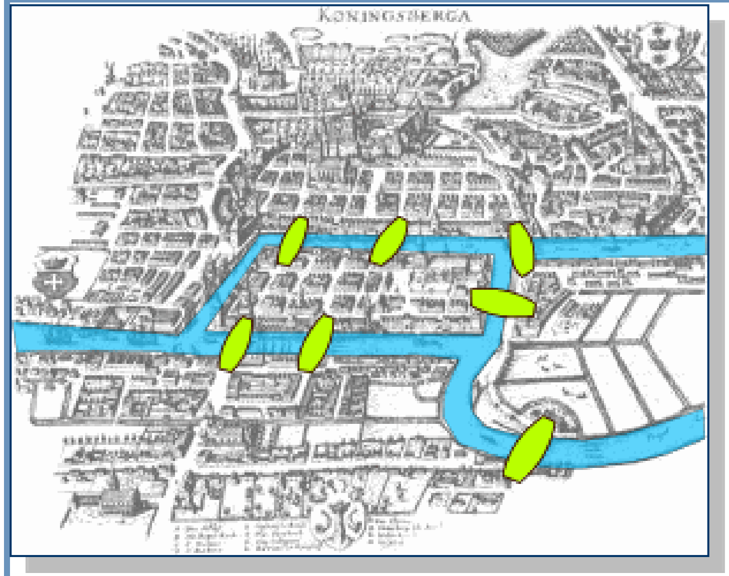
## 基于 Link-Path 的建模方法减少约束和决策变量了吗?

- #流守恒约束 $N(N+M)(N-1)$ ⇨ #需求量约束 $N(N-1)$
- #link-flow 变量 $MN(N-1)$ ⇨ # path 变量 $|P|$

# Conclusion

## What we have done:

- 我们基于 Node-Link 建立了可变需求矩阵的网络模型
- 我们探究了如何求解 Linear Semi-infinite Programming 问题
- 我们运用了对偶的手段，将两个相互依赖的 LP 问题合为一体
- 我们重新将我们的问题建成为了 Link-Path 模型
- 我们简单理解了 Column-Generation 算法如何提升 Simplex 算法




# Thanks for your time

一种对未知流量的域内路由算法

汇报人：黄卓彬

指导老师：王雄副教授