第2讲：锥规划及其在能源系统优化中的应用．Part II

许　寅

北京交通大学电气工程学院

往年教学录屏：https://www.bilibili.com/video/BV1F7411o7Jx/

思考题

如何用向量或矩阵形式表达以下约束条件？

$$a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = b$$

优化变量、目标函数与约束条件的表示形式

1. 标量形式：优化变量x_1, x_2, x_3, x_4满足约束条件

$$a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = b$$

2. 向量形式：优化变量$x = [x_1, x_2, x_3, x_4]^T$，参数$a = [a_1, a_2, a_3, a_4]^T$，则上述约束条件可表示为

向量内积 $a^T x = b$

3. 矩阵形式：将优化变量和参数组织成矩阵形式

$$X = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \qquad A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$$

矩阵内积：对应元素相乘后求和

$$\mathbf{tr}(A^T X) = b$$

其中，$\mathbf{tr}(\cdot)$表示一个矩阵的迹，即矩阵对角元之和

半定锥 (Positive semidefinite cone)

Set of **symmetric positive semidefinite matrices:**

$$\mathcal{S}_+^n = \{X \in \mathcal{S}^n | X \succcurlyeq 0\},$$

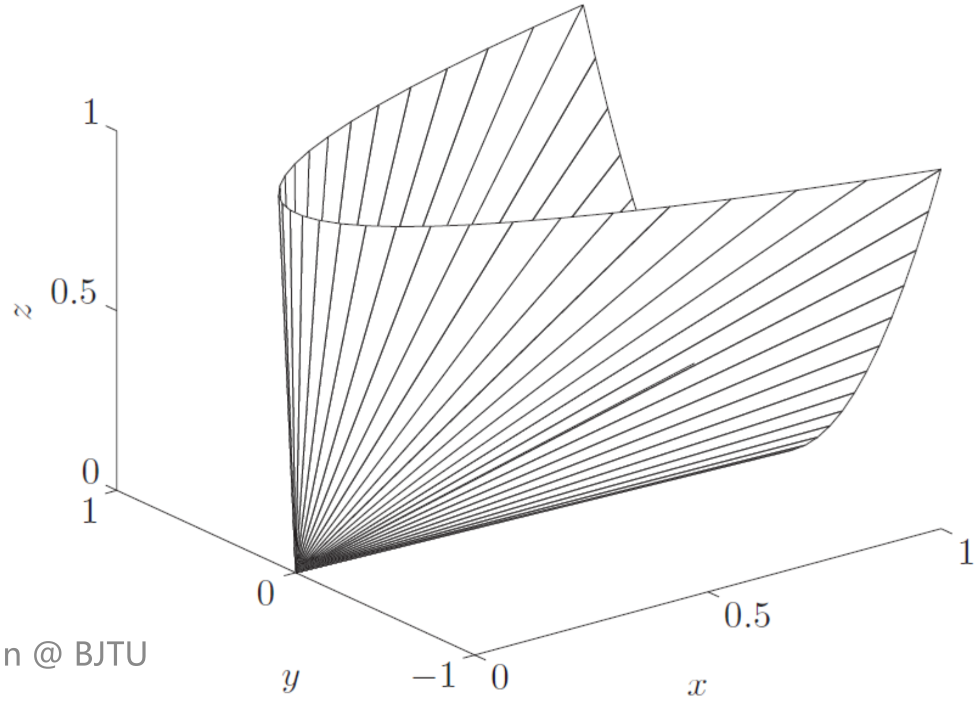
where $\mathcal{S}^n = \{X \in \mathcal{R}^{n\times n} | X = X^T\}$. The set $\mathcal{S}_+^n$ is a convex cone.

$$X = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \in \mathcal{S}_+^2$$

$$\Updownarrow$$

$$x \geq 0, z \geq 0, xz \geq y^2$$

旋转的二阶锥



半定规划 (Semidefinite programming, SDP)

SDP的标准形式

$$
\begin{array}{ll}
\text{minimize} & \mathbf{tr}(CX) \quad \text{线性目标函数} \\
\text{subject to} & \mathbf{tr}(A_iX) = b_i, i = 1, \dots, p \quad \text{线性等式约束} \\
& X \succcurlyeq 0 \quad \text{半定约束}
\end{array}
$$

其中 $C, A_1, \dots, A_p \in \mathcal{S}^n$

例题

将以下半定规划模型转换为标量表示形式。

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{tr}(CX) \\
\text{subject to} \quad & \mathbf{tr}(A_1X) = b_1 \\
& \mathbf{tr}(A_2X) = b_2 \\
& X \succcurlyeq 0
\end{aligned}
$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{12} & x_{22} & x_{23} \\ x_{13} & x_{23} & x_{33} \end{bmatrix}$$

共6个优化变量

$$A_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{bmatrix} \quad b_1 = 11 \text{ and } b_2 = 19$$

解答

$$\text{minimize} \quad x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 7x_{33}$$

$$\text{subject to} \quad x_{11} + 2x_{13} + 3x_{22} + 14x_{23} + 5x_{33} = 11$$

$$4x_{12} + 16x_{13} + 6x_{22} + 4x_{33} = 19$$

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{12} & x_{22} & x_{23} \\ x_{13} & x_{23} & x_{33} \end{bmatrix} \succcurlyeq 0$$

- 半定约束等价于该矩阵的各阶主子式不小于零
- 一共多少个不等式约束？

解答

对称矩阵A半正定的充分必要条件是A的各阶主子式都不小于零

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{12} & x_{22} & x_{23} \\ x_{13} & x_{23} & x_{33} \end{bmatrix} \succcurlyeq 0 \quad \Leftrightarrow \quad x_{11} \geq 0, x_{22} \geq 0, x_{33} \geq 0$$

$$\begin{vmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{vmatrix} = x_{11}x_{22} - x_{12}^2 \geq 0 \quad x_{11}x_{33} - x_{13}^2 \geq 0 \quad x_{22}x_{33} - x_{23}^2 \geq 0$$

$$\begin{vmatrix} x_{11} & x_{12} & x_{13} \\ x_{12} & x_{22} & x_{23} \\ x_{13} & x_{23} & x_{33} \end{vmatrix} = x_{11}x_{22}x_{33} + 2x_{12}x_{23}x_{13} - x_{11}x_{23}^2 - x_{22}x_{13}^2 - x_{33}x_{12}^2 \geq 0$$

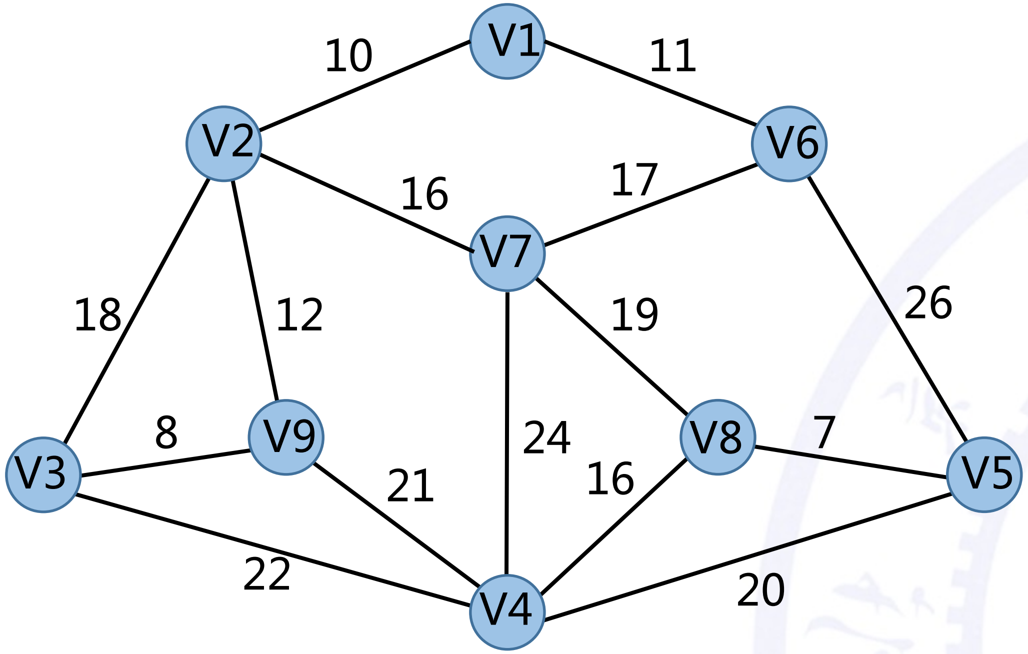
思考题

采用n阶实对称矩阵X表示优化变量，请问：
(1) 共有多少个优化变量（标量）？
(2) 半定约束$X \succcurlyeq 0$可等价转换为几个标量形式的不等式约束？

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{1n} & \cdots & x_{nn} \end{pmatrix} \succcurlyeq 0$$

应用1：最大割问题 Maximum Cut

- 已知：无向图 $G = (\mathcal{N}, \mathcal{E})$，边$(i,j) \in \mathcal{E}$的权重为$\omega_{ij} \geq 0$且$\omega_{ij} = \omega_{ji}$
- 求：节点子集 $\mathcal{S} \subseteq \mathcal{N}$，使得 $\sum_{i\in\mathcal{S},j\in\bar{\mathcal{S}}} \omega_{ij}$最大，其中 $\bar{\mathcal{S}} = \mathcal{N} \backslash \mathcal{S}$



最大割问题建模

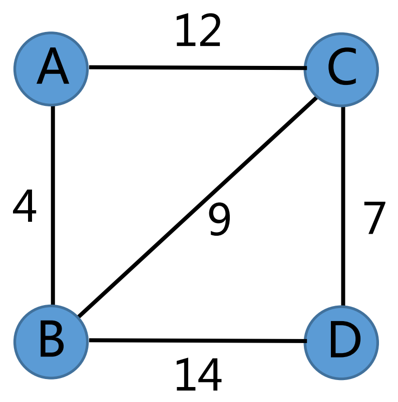
Let $x_i = 1$ if $i \in \mathcal{S}$ and $x_i = -1$ if $i \in \bar{\mathcal{S}}$.

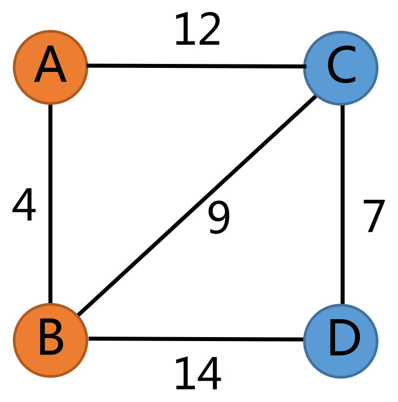
$$\text{maximize} \quad \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}(1 - x_i x_j)$$

$$\text{s.t.} \quad x_j \in \{-1,1\}, \qquad j = 1,\ldots,n$$

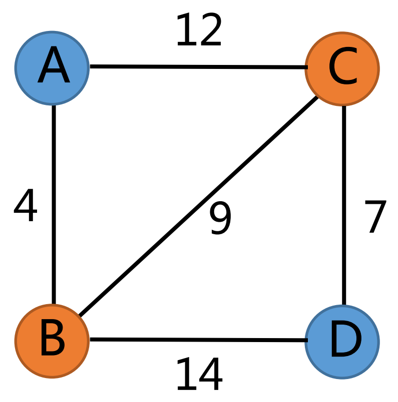
组合优化问题：**组合爆炸！**一共有多少种可能性？

4节点最大割问题分析（穷举法）

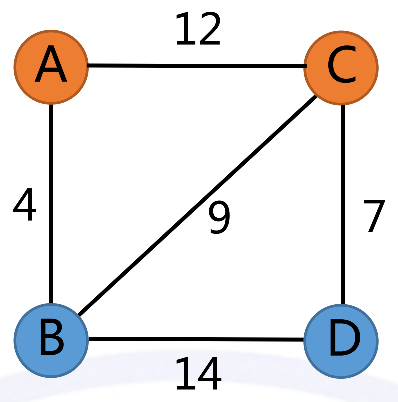




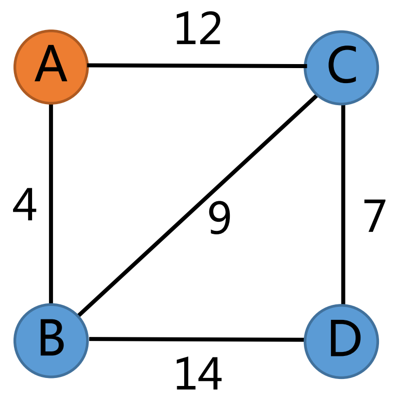
AB&CD:12+9+14=35



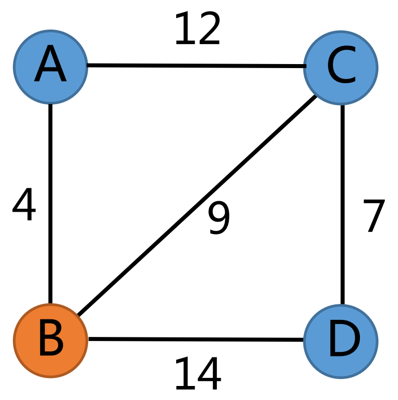
AD&BC:12+4+14+7=37



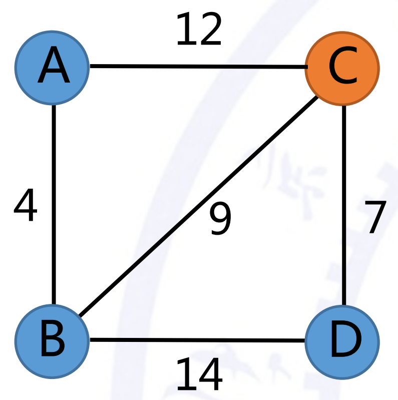
AC&BD:4+9+7=20



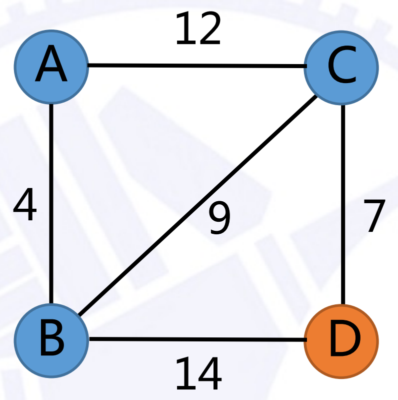
A&BCD:12+4=16



B&ACD:4+9+14=27



C&ABD:12+9+7=28



D&ABC:7+14=21

变量替换

- 注意到变量x_i始终以乘积形式$x_i x_j$出现，令$Y = xx^T$，则$Y_{ij} = x_i x_j$
- 定义矩阵 W，其第 i 行第 j 列的元素为 ω_{ij}，对角元为0

$$
\begin{aligned}
\text{maximize} \quad & \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}\left(1 - x_i x_j\right) \\
\text{s.t.} \quad & x_j \in \{-1,1\}, \qquad j = 1,\ldots,n
\end{aligned}
$$

$\Rightarrow$

$$
\begin{aligned}
\text{maximize} \quad & \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij} - \frac{1}{4}\mathbf{tr}(WY) \\
\text{s.t.} \quad & Y_{jj} = 1, \qquad j = 1,\ldots,n \\
& Y = xx^T
\end{aligned}
$$

松弛：去掉秩1约束

$$Y = xx^T \quad \Longleftrightarrow \quad Y \succcurlyeq 0 \text{ and } \mathbf{rank}(Y) = 1$$

$$\text{maximize} \quad \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij} - \frac{1}{4}\mathbf{tr}(WY)$$

$$\text{s.t.} \quad Y_{jj} = 1, \qquad j = 1,\ldots,n$$

$$Y = xx^T$$

思考：求解得到的Y矩阵非对角元可能是小数，如何得到原优化变量x的取值？

$$\Rightarrow$$

$$\text{maximize} \quad \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij} - \frac{1}{4}\mathbf{tr}(WY)$$

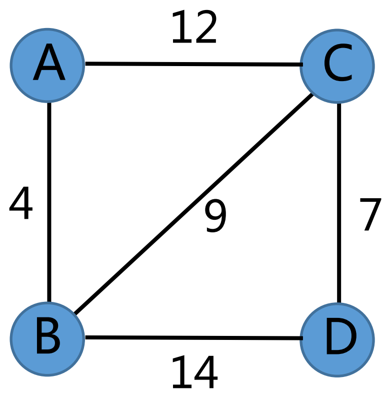
$$\text{s.t.} \quad Y_{jj} = 1, \qquad j = 1,\ldots,n$$

$$Y \succcurlyeq 0$$

$$\cancel{\mathbf{rank}(Y) = 1}$$ 秩1松弛

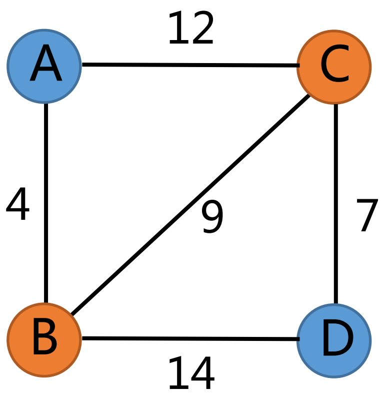
$$\mathbf{0.87856\,\mathit{RELAX} \le \mathit{MAXCUT} \le \mathit{RELAX}}$$

四节点算例



$$W=\begin{bmatrix}0&4&12&0\\4&0&9&14\\12&9&0&7\\0&14&7&0\end{bmatrix}$$

SDP松弛模型求解结果：37.2364136672298



AD&BC:12+4+14+7=37

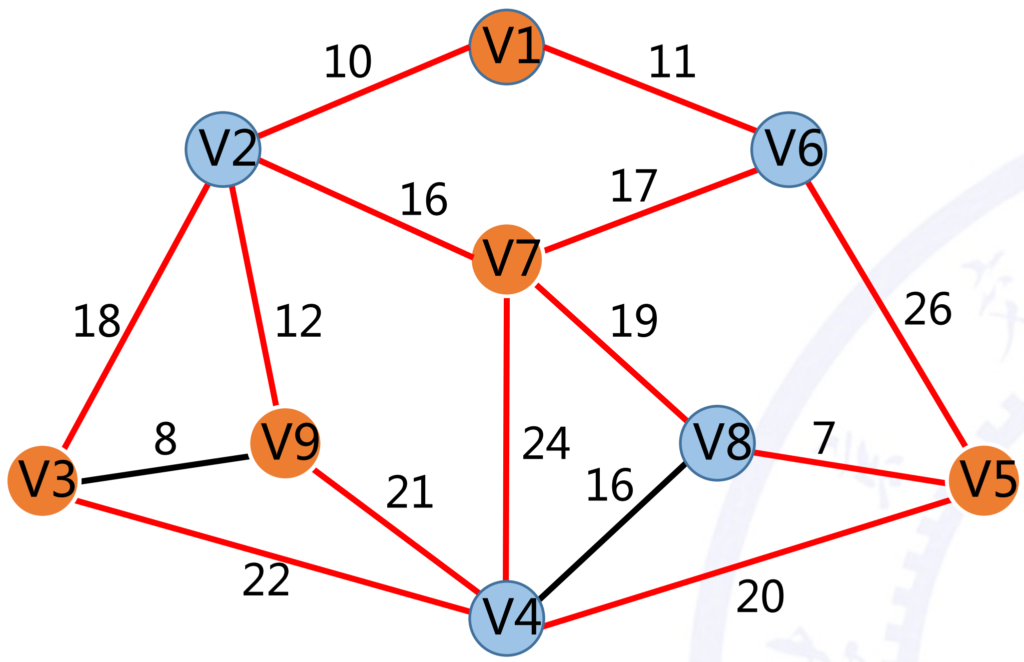
$$Y=\begin{bmatrix}1.0&-0.584843&-0.962764&0.844561\\-0.584843&1.0&0.343777&-0.928271\\-0.962764&0.343777&1.0&-0.668356\\0.844561&-0.928271&-0.668356&1.0\end{bmatrix}$$

取整：非对接元大于0则取1，小于0则取-1

对于其他算例未必可行

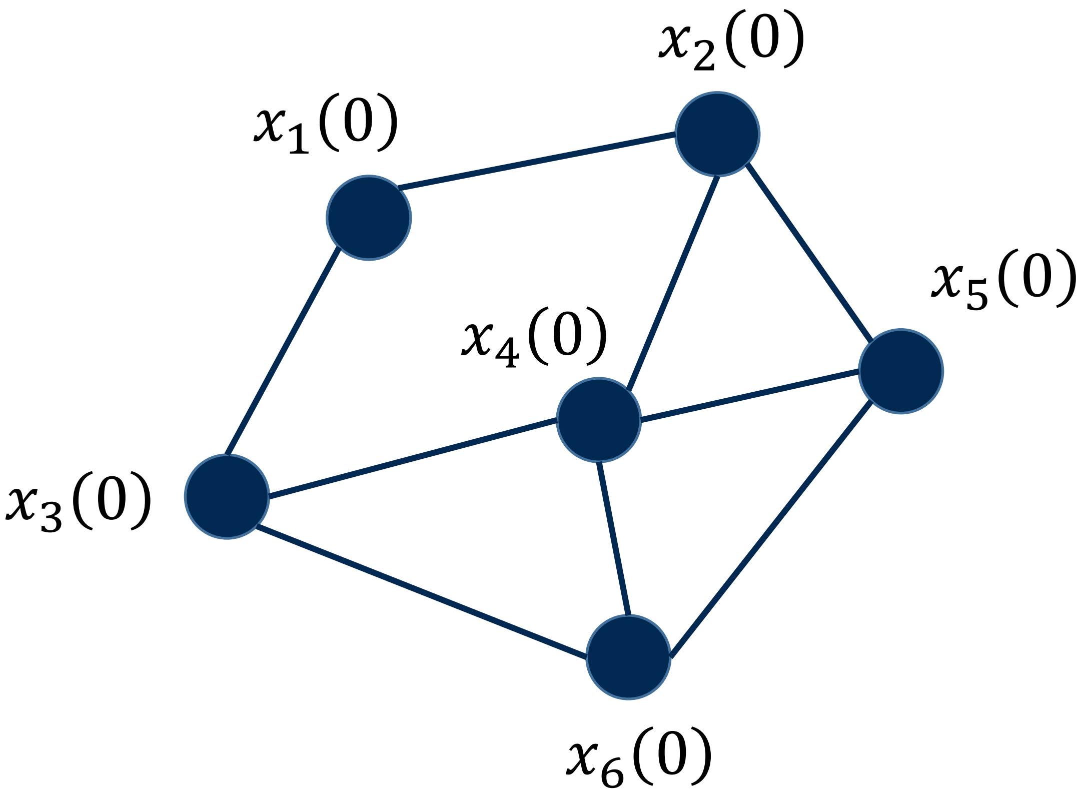
9节点图的最大割求解结果

SDP模型求解结果：目标函数值为223.000000053779

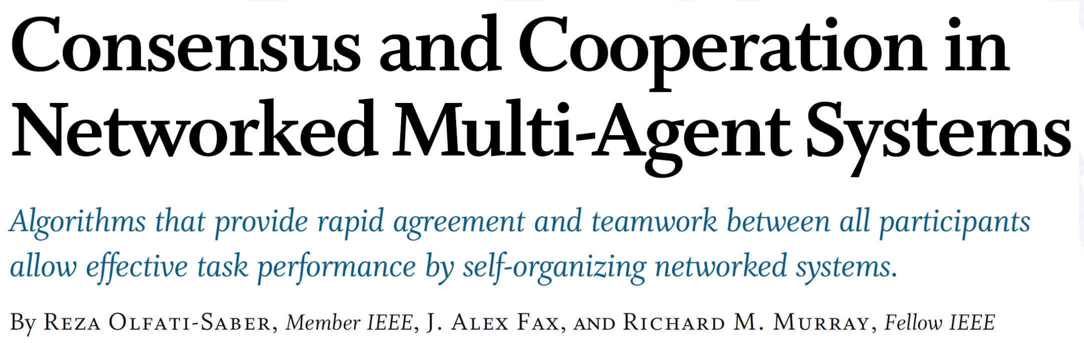


应用2：分布式控制的最优参数设计

Average Consensus in Networked Multi-Agent Systems

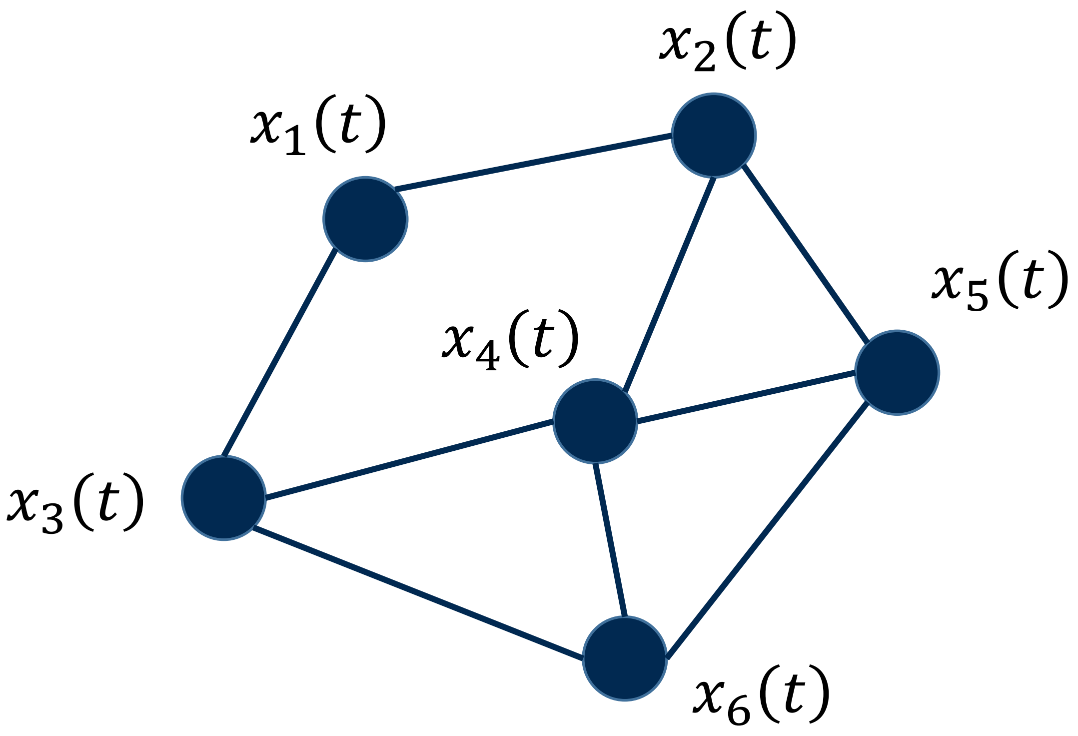


- 每一个节点有一个初始状态$x_i(0) \in \mathcal{R}$
- 每一个节点可以与其相邻节点通信
- 如何通过**分布式算法**计算所有节点初始状态的平均值？



分布式控制最优参数设计的问题描述

Average Consensus in Networked Multi-Agent Systems



Distributed Linear Iteration

$$x_i(t+1) = W_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} W_{ij}x_j(t)$$

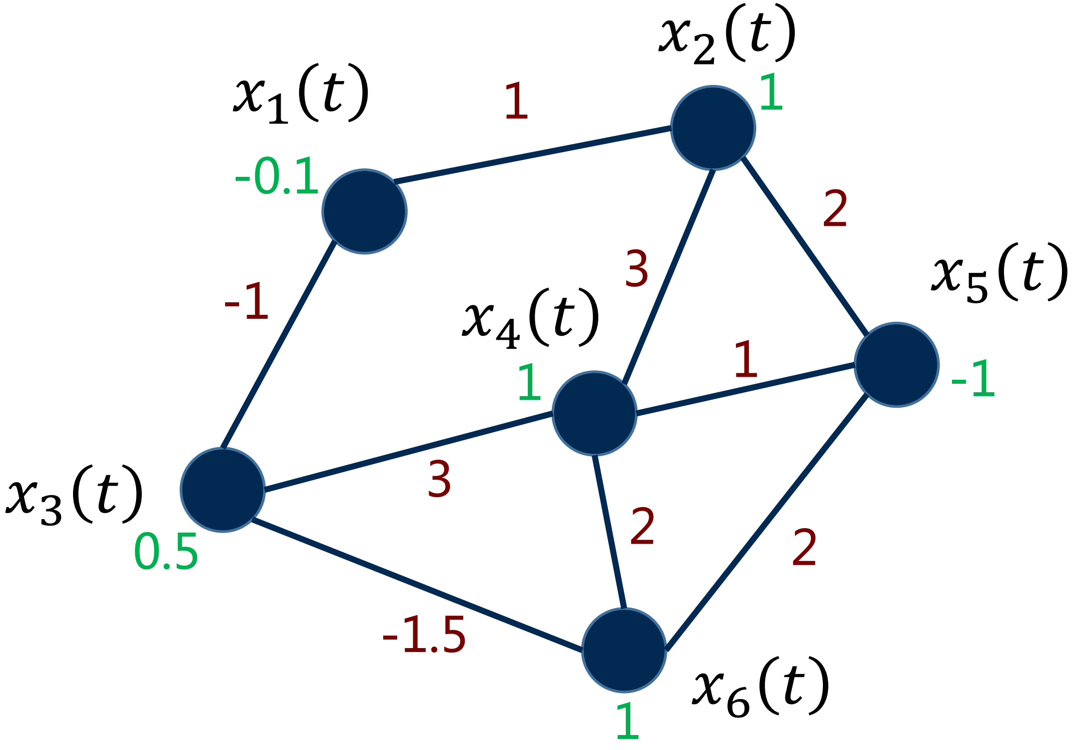
$$\Downarrow$$

$$x(t+1) = Wx(t)$$

$$W \in \mathcal{L} = \{W \in \mathcal{S}^n | W_{ij} = 0 \; if \; (i,j) \notin \mathcal{E}\}$$

问题：如何设计矩阵W的参数，使得该分布式算法收敛速度最快（收敛所需迭代次数最小）？

矩阵W举例



$$x_i(t+1) = W_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} W_{ij}x_j(t)$$

$$x(t+1) = Wx(t)$$

$$W = \begin{bmatrix} -0.1 & 1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 3 & 2 & 0 \\ -1 & 0 & 0.5 & 3 & 0 & -1.5 \\ 0 & 3 & 3 & 1 & 1 & 2 \\ 0 & 2 & 0 & 1 & -1 & 2 \\ 0 & 0 & -1.5 & 2 & 2 & 1 \end{bmatrix}$$

矩阵W须满足的条件

$x(t)=[x_1(t),\dots,x_n(t)]^T$ 已知 $x(0)=[x_1(0),\dots,x_n(0)]^T$

分布式协议 (protocol): $x(t+1)=Wx(t)$

分布式协议收敛至平均值: 当$t\to\infty$时，

$$x_1(t)=\cdots=x_n(t)=\frac{x_1(0)+\cdots+x_n(0)}{n}$$ 即 $$\lim_{t\to\infty}x(t)=\left(\frac{\mathbf{1}^T x(0)}{n}\right)\mathbf{1}=\left(\frac{\mathbf{1}\mathbf{1}^T}{n}\right)x(0)$$

条件1：$x=\mathbf{1}$ 是函数$f(x)=Wx$的不动点，即$W\mathbf{1}=\mathbf{1}$

条件2：迭代过程中各agent状态$x_i(t)$的平均值保持不变，即$\mathbf{1}^TW=\mathbf{1}^T$

（说明：由于我们假设 W 为对称矩阵，条件1与条件2只要满足其中1项，另1项自然满足）

矩阵W须满足的条件

- 如何能确保经过一定次数迭代后，$x_1(t) = \cdots = x_n(t)$ ？

思路：

状态的初始值 $x(0) = [x_1(0), \dots, x_n(0)]^T$

我们希望 $\lim_{t \to \infty} x(t) = \left(\frac{\mathbf{1}\mathbf{1}^T}{n}\right) x(0)$

定义 $\Delta x(t) = x(t) - \left(\frac{\mathbf{1}\mathbf{1}^T}{n}\right) x(0)$

我们希望 $\lim_{t \to \infty} \Delta x(t) = \mathbf{0}$

【数学知识回顾】特征值分解

实对称矩阵的特征值分解

Suppose $A \in \mathcal{S}^n$, i.e., A is a real symmetric $n \times n$ matrix. Then A can be factored as

$$A = Q\Lambda Q^T$$

where $Q \in \mathcal{R}^{n \times n}$ is orthogonal, i.e., satisfies $Q^T Q = I$, and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$. The (real) numbers $\lambda_1 \geq \cdots \geq \lambda_n$ are the eigenvalues of A. The columns of Q form an orthonormal set of eigenvectors of A.

$$A = Q\Lambda Q^T = \lambda_1 q_1 q_1^T + \cdots + \lambda_n q_n q_n^T$$

矩阵W须满足的条件

条件1重新表述：$\lambda_1 = 1$是矩阵W的特征值，$q_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ 是相应的单位特征向量，即$Wq_1 = 1q_1$ (等价于$W\mathbf{1} = \mathbf{1}$)

对矩阵W进行特征值分解：$W = Q\Lambda Q^T = \frac{\mathbf{1}\mathbf{1}^T}{n} + \lambda_2 q_2 q_2^T + \cdots + \lambda_n q_n q_n^T$

$$\Delta x(1) = x(1) - \frac{\mathbf{1}\mathbf{1}^T}{n}x(0) = \left(W - \frac{\mathbf{1}\mathbf{1}^T}{n}\right)x(0) = (\lambda_2 q_2 q_2^T + \cdots + \lambda_n q_n q_n^T)x(0)$$

$$\Delta x(2) = x(2) - \frac{\mathbf{1}\mathbf{1}^T}{n}x(0) = \left(W^2 - \frac{\mathbf{1}\mathbf{1}^T}{n}\right)x(0) = (\lambda_2^2 q_2 q_2^T + \cdots + \lambda_n^2 q_n q_n^T)x(0)$$

以此类推，$\boxed{\Delta x(t) = (\lambda_2^t q_2 q_2^T + \cdots + \lambda_n^t q_n q_n^T)x(0)}$

矩阵W须满足的条件

$$\boxed{\Delta x(t) = (\lambda_2^t q_2 q_2^T + \cdots + \lambda_n^t q_n q_n^T) x(0)}$$

对于任意$x(0)$， $\lim_{t\to\infty} \Delta x(t) = \mathbf{0}$的充要条件是

$$\lim_{t\to\infty} (\lambda_2^t q_2 q_2^T + \cdots + \lambda_n^t q_n q_n^T) = \mathbf{0}$$

$$\Rightarrow \quad |\lambda_2| < 1, \ldots, |\lambda_n| < 1$$

$$\Leftrightarrow \quad \rho\left(W - \frac{\mathbf{1}\mathbf{1}^T}{n}\right) = \max\{|\lambda_2|, \ldots, |\lambda_n|\} < 1$$

谱半径决定了是否收敛及收敛速度

确定最优参数

$$\mathcal{L} = \{W \in \mathcal{S}^n | W_{ij} = 0 \; if \; (i,j) \notin \mathcal{E}\}$$

$$\begin{aligned} &\text{minimize} && \rho\left(W - \frac{\mathbf{1}\mathbf{1}^T}{n}\right) \\ &\text{subject to} && W \in \mathcal{L} \quad W\mathbf{1} = \mathbf{1} \quad \cancel{\rho\left(W - \frac{\mathbf{1}\mathbf{1}^T}{n}\right) < 1} \end{aligned}$$

实对称矩阵A的特征值均不小于(不大于) s 的充分必要条件是 $A - sI \succcurlyeq 0$ $(A - sI \preccurlyeq 0)$

因此，上述优化问题可转化为如下SDP问题：

$$\begin{aligned} &\text{minimize} && s \\ &\text{subject to} && -sI \preccurlyeq W - \frac{\mathbf{1}\mathbf{1}^T}{n} \preccurlyeq sI \quad \text{半定约束} \\ &&& W \in \mathcal{L} \quad W\mathbf{1} = \mathbf{1} \end{aligned}$$

半定规划(SDP)求解器

- **商业软件**：
  - **Mosek**: (MI)LP, (MI)SOCP, **SDP** https://www.mosek.com

- **开源软件**：
  - **COSMO.jl**: LP, QP, SOCP, **SDP** https://github.com/oxfordcontrol/COSMO.jl
  - **SCS**: LP, SOCP, **SDP** https://github.com/cvxgrp/scs
  - **CSDP**: LP, **SDP** https://github.com/coin-or/Csdp

更多关于JuMP支持的求解器的信息参见官方说明文档：
https://jump.dev/JuMP.jl/stable/installation/#Supported-solvers

在Julia/JuMP中构建半定约束

- **例**：半定约束可采用以下方式构建：

```
julia> @variable(model, X[1:2, 1:2])
2×2 Matrix{VariableRef}:
 X[1,1]  X[1,2]
 X[2,1]  X[2,2]

julia> @constraint(model, X >= 0, PSDCone())
[X[1,1]  X[1,2];
 X[2,1]  X[2,2]] ∈ PSDCone()
```

```
julia> Y = [1 2; 2 1]
2×2 Matrix{Int64}:
 1  2
 2  1

julia> @constraint(model, X >= Y, PSDCone())
[X[1,1] - 1  X[1,2] - 2;
 X[2,1] - 2  X[2,2] - 1] ∈ PSDCone()
```

如何定义对称矩阵变量？了解关于JuMP半定约束的更多信息：

https://jump.dev/JuMP.jl/stable/manual/constraints/#Semidefinite-constraints

直接构建半定矩阵变量

- 方法1：

```
julia> @variable(model, x[1:2, 1:2], PSD)
2×2 LinearAlgebra.Symmetric{VariableRef, Matrix{VariableRef}}:
 x[1,1]  x[1,2]
 x[1,2]  x[2,2]
```

https://jump.dev/JuMP.jl/stable/manual/variables/#Semidefinite-variables

- 方法2：

```
julia> @variable(model, x[1:2, 1:2] in PSDCone())
2×2 LinearAlgebra.Symmetric{VariableRef, Matrix{VariableRef}}:
 x[1,1]  x[1,2]
 x[1,2]  x[2,2]
```

https://jump.dev/JuMP.jl/stable/manual/variables/#Example:-positive-semidefinite-variables

推荐采用直接构建半定矩阵变量的方式，对于部分求解器计算效率更高

范例程序

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{tr}(CX) \\
\text{subject to} \quad & \mathbf{tr}(AX) = b \\
& X \succcurlyeq 0
\end{aligned}
$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

$$A = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$b = 9$$

```julia
using JuMP,CSDP,LinearAlgebra
C = [1 0;0 4]
A = [5 6;7 8]
b = 9
model=Model(CSDP.Optimizer)
# 方法一：直接建立半定矩阵变量
@variable(model,X[1:2,1:2],PSD)
# 方法二：利用半定锥构建半定矩阵变量
# @variable(model,X[1:2,1:2] in PSDCone())
# 方法三：先定义对称矩阵变量，再增加半定约束
# @variable(model,X[1:2,1:2],Symmetric)
# @constraint(model,X>=0,PSDCone())
@objective(model,Min,dot(C,X))
@constraint(model,dot(A,X)==b)
print(model)
optimize!(model)
println("程序终止状态：", termination_status(model))
println("原问题状态：", primal_status(model))
println("最优值：",objective_value(model))
println("最优解：X = ",value.(X))
```

作业1

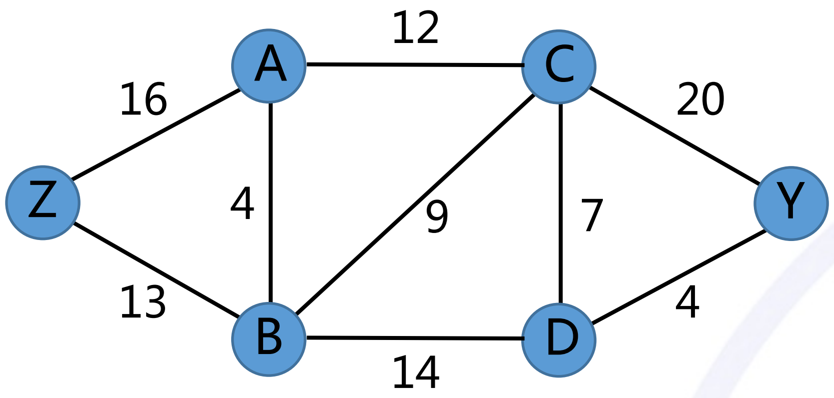
编程求解以下SDP问题

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{tr}(CX) \\
\text{subject to} \quad & \mathbf{tr}(A_1 X) = b_1 \\
& \mathbf{tr}(A_2 X) = b_2 \\
& X \succcurlyeq 0
\end{aligned}
$$

$$
A_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{bmatrix} \quad b_1 = 11 \text{ and } b_2 = 19
$$

作业2

- 求下图的最大割



作业3（选做）

- 给定如下多代理网络，为使得平均一致性算法收敛速度最快，请确定最优的参数W，并将结果标注在图上（W_{ii}标注在节点上，W_{ij}标注在边上）