CO327 Deterministic OR Models (2021-Spring)

MAXCUT, quadratic program and semidefinite program

Instructor: Andersen Ang
Guest lecturer: Mariia Sobchuk msobchuk@uwaterloo.ca

Combinatorics and Optimization, U.Waterloo, Canada
msxang@uwaterloo.ca, where $\mathbf{x} = \lfloor \pi \rfloor$
Homepage: angms.science

First draft: April 3, 2021 Last update: ~~May 28~~, 2021
June 20

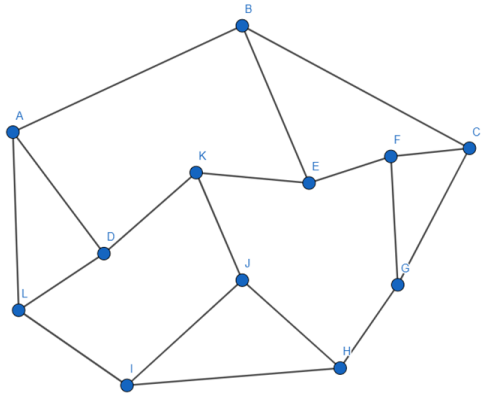
MAXCUT

- Undirected graph $G(V, E)$ with no self-loops
  - Vertex set $V = [n] \coloneqq \{1, 2, \ldots, n\}$
  - Edge set $E \subset \left\{ \{i, j\} \;\middle|\; i, j \in V, i \neq j \right\}$
- For of a subset S of V, the capacity of S is the number of edges connecting a node in S to a node not in S
- MAXCUT problem: find $S \subset V$ with maximum capacity
- Application of MAXCUT: physics, electronic engineering, social network modelling.

MAXCUT, descried in graph language

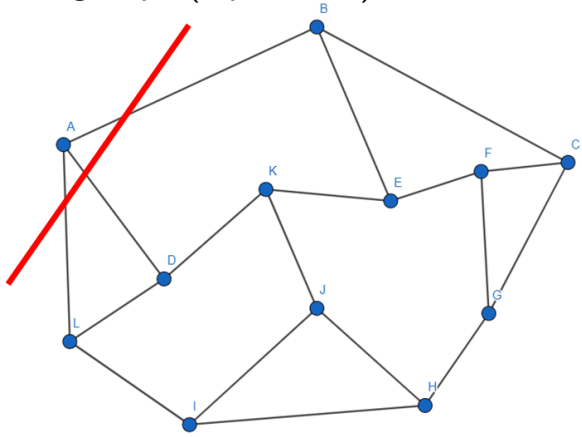
"Given a graph, find a **bipartition** of the vertex set that cuts as many edges as possible."

Example



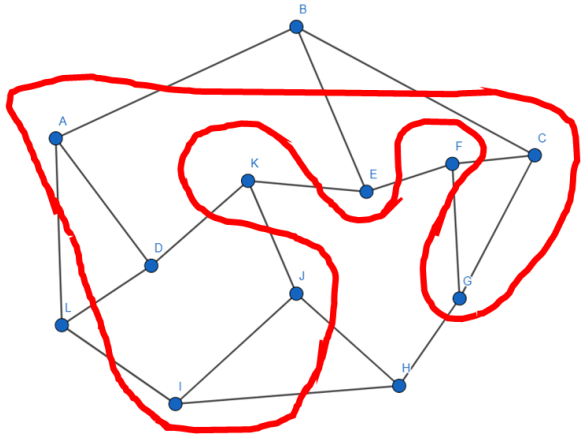
- $V = \Big\{A, B, C, D, \ldots, L\Big\}$
- $E = \Big\{(A, B), (A, D), (A, L), (B, A), (B, C), (B, E), \ldots\Big\}$

A cut split V into two groups (bipartition)



- $V = \{A, B, C, D, \ldots, L\}$
- $S = \{A\} \subset V$ or $S = \{B, C, D, \ldots, L\} \subset V$
- Capacity of $S = 3$

MAXCUT

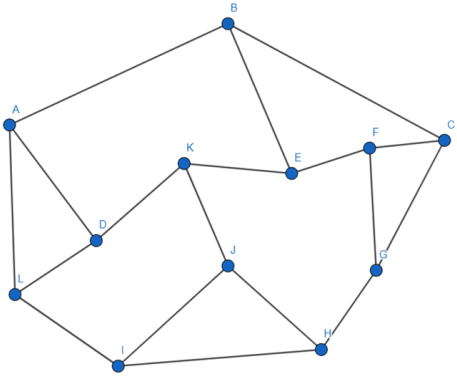


- $S = \{A, C, D, E, G, I, J\} \subset V$ or $S = \{B, F, H, K, L\} \subset V$
- Capacity of $S = 15$

Problem Formulation: adjacency matrix (the data)

A graph is defined by its adjacency matrix

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{else} \end{cases}$$



$$\mathbf{A} = \begin{bmatrix} & A & B & C & D & \dots \\ A & 0 & 1 & 0 & 1 & \dots \\ B & 1 & 0 & 1 & 0 & \dots \\ C & 0 & 1 & 0 & 0 & \dots \\ D & 1 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Integer matrix

- The adjacency matrix $\mathbf{A}$
  - $\mathbf{A} \in \mathbb{N}^{n \times n}$: it is a square integer matrix.
  - $\mathbf{A} = \mathbf{A}^\top$: it is symmetric for undirected graph.
    Linear algebra: $\mathbf{A}$ has real eigenvalues (and real eigenvectors).
  - $\mathrm{diag}(\mathbf{A}) = \mathbf{0}$: its diagonal is zero for graph with no self-loop.

- Definition (Unimodular) A square integer matrix with determinant $+1$ or -1 is called unimodular.

- Total unimodular (TU): all non-singular submatrix is unimodular.

- Property: if $\mathbf{A}$ is TU and $\mathbf{b}$ is integral, then all vertex sol. of $\mathbf{Ax} \geq \mathbf{b}$ is integral.

Problem Formulation: decision variable (the cut)

- Decision variable: x_i
- A cut S is defined by a vector $\mathbf{x} \in \mathbb{R}^n$

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{else} \end{cases} \qquad (*)$$

- If edge (i, j) across the cut
  - Vertex i and vertex j are separated by a cut
  - $\iff$ one of them is in S and the other one is outside S
  - $\iff$ x_i and x_j have opposite sign according to (*)
  - $\iff$ $x_i x_j = -1$
- Therefore,

$$(i, j) \text{ is a cut} \iff 1 - x_i x_j = 2.$$

- What if edge (i, j) across the cut? What if (i, j) is not an edge?

Problem Formulation: cost function (capacity)

- Capacity of the cut

$$c(x) = \sum_{(i,j)\in E} (1 - x_i x_j) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}(1 - x_i x_j)$$

- Understand it:
  - If (i, j) is a cut: $1 - x_i x_j \neq 0$
  - If (i, j) is not a cut: $1 - x_i x_j = 0$
  - If (i, j) are adjacent: $A_{ij} \neq 0$
  - If (i, j) are not adjacent: $A_{ij} = 0$

- Goal: ~~minimize~~ maximize the capacity

$$\max_{\mathbf{x}} \sum_{i,j} A_{ij}(1 - x_i x_j).$$

Convention

$$\max_{\mathbf{x}} \frac{1}{2} \sum_{i,j} A_{ij}(1 - x_i x_j).$$

Quadratic integer program

$$\max_{\mathbf{x}} \ \frac{1}{2}\sum_{i,j} A_{ij}(1 - x_i x_j) \ \text{ s.t. } \ x_i \in \{-1, +1\}, \ \forall i.$$

- Equivalent expressions

$$\max_{\mathbf{x}} \frac{1}{4}\sum_{i,j} A_{ij}(x_i - x_j)^2 \ \text{ s.t. } \ x_i \in \{-1, +1\}, \ \forall i$$

  Why: $(x_i - x_j)^2 = \underbrace{x_i^2 + x_j^2}_{=1+1} - 2x_i x_j = 2(1 - x_i x_j)$.

- Can model the constraints with quadratic equations:

$$\max_{\mathbf{x}} \ \frac{1}{4}\sum_{i,j} A_{ij}(x_i - x_j)^2 \ \text{ s.t. } \ x_i^2 - 1 = 0, \ \forall i.$$

- Recall LIP are in general NP-hard. Same for QIP.

Vector quadratic form expression

- Linear algebra fact: the quadratic function $\frac{1}{4}\sum_{i,j} A_{ij}(x_i - x_j)^2$ can be expressed as $\mathbf{x}^\top \mathbf{Q}\mathbf{x}$ where $\mathbf{Q}$ is a square symmetric matrix.

$$
\begin{aligned}
A_{ij}(x_i - x_j)^2 &= A_{ij}(x_i^2 - 2x_i x_j + x_j^2) \\
&= A_{ij}x_i^2 - A_{ij}2x_i x_j + A_{ij}x_j^2 && \text{expand} \\
&= A_{ij}x_i^2 - A_{ij}x_i x_j - A_{ij}x_i x_j + A_{ij}x_j^2 \\
&= x_i A_{ij} x_i - x_i A_{ij} x_j - x_j A_{ij} x_i + x_j A_{ij} x_j \\
&= \begin{bmatrix} \vdots \\ x_i \\ x_j \\ \vdots \end{bmatrix}^\top \begin{bmatrix} \ddots & \vdots & \vdots & \vdots \\ \cdots & A_{ij} & -A_{ij} & \cdots \\ \cdots & -A_{ij} & A_{ij} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ x_i \\ x_j \\ \vdots \end{bmatrix}
\end{aligned}
$$

  and similar for all other pairs.

- We have $\max_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x}$ s.t. $x_i^2 - 1 = 0,\ \forall i$.

QP over hypercube

MAXCUT as a quadratic optimization problem over hypercube

$$\max_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} \text{ s.t. } x_i \in \{-1, +1\}, \ \forall i.$$

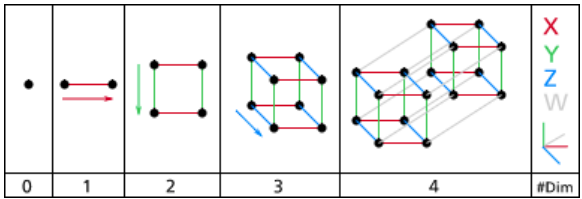


Figure: Hypercube lattice. Image source: wikipedia.

Examples

- 1-dimension: $\max_x \frac{1}{2}ax^2 \text{ s.t. } x \in \{-1, +1\}$.
- 2-dimension: $\max_{x,y} \frac{1}{2}\begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{ s.t. } x, y \in \{-1, +1\}$.

General QP

$$\min_{\mathbf{x}\in\mathcal{C}} \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} - \mathbf{p}^\top \mathbf{x} + c$$

- $\mathbf{x} \in \mathbb{R}^n$: vector variable
- $\mathbf{Q} \in \mathbb{R}^{n\times n}$: a matrix (it can be any matrix in general but usually we assume it is positive semidefinite and symmetric)
- $\mathbf{p} \in \mathbb{R}^n$ a vector
- $c \in \mathbb{R}$ a scalar
- $\mathcal{C}$ the constraint set. If $\mathcal{C} = \mathbb{Z}$ we have a QIP.
- How to solve QP: either by a general purpose solver or write your own code.

Yet another formulation ... (1/3)

- MAXCUT expressed using adjacency matrix:

$$\max_{\mathbf{x}} \frac{1}{2} \sum_{i,j} A_{ij}(1 - x_i x_j) \text{ s.t. } x_i^2 = 1, \ \forall i.$$

- Introduce a new variable $\mathbf{Y} \in \mathbb{R}^{n \times n}$, a matrix defined as $Y_{ij} = x_i x_j$:

$$\sum_{i,j} A_{ij}(1 - x_i x_j) = \sum_{i,j} \Big(A_{ij} - A_{ij} Y_{ij} \Big) = \underbrace{\sum_{i,j} A_{ij}}_{\text{constant}} - \sum_{i,j} A_{ij} Y_{ij}$$

- $x_i^2 = 1$ translates to $Y_{ii} = 1$.

- Ignore the constant and the $\frac{1}{2}$ gives

$$\max_{\mathbf{x}} - \sum_{i,j} A_{ij} Y_{ij} \text{ s.t. } \mathbf{Y} = \mathbf{x}\mathbf{x}^\top, \ Y_{ii} = 1 \ \forall i.$$

Yet another formulation ... (2/3)

- The term $\sum_{i,j} A_{ij} Y_{ij}$ is actually the inner product between $\mathbf{A}$ and $\mathbf{Y}$.
  (Recall vector inner product: $\mathbf{a}^\top \mathbf{y} = \sum_i a_i y_i$.)
- A linear Algebra fact: $\sum_{i,j} A_{ij} Y_{ij} = \mathrm{Tr}\left(\mathbf{A}^\top \mathbf{Y}\right)$
- Using inner product notation:

$$\max_{\mathbf{x}} -\langle \mathbf{A}, \mathbf{Y} \rangle \ \text{ s.t. } \ \mathbf{Y} = \mathbf{x}\mathbf{x}^\top, \ Y_{ii} = 1 \ \forall i.$$

Yet another formulation ... (3/3)

- By construct, $\mathbf{Y}$ is a rank-1 matrix

$$\mathbf{Y} = \mathbf{x}\mathbf{x}^\top = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots \end{bmatrix} = \begin{bmatrix} x_1x_1 & x_1x_2 & \cdots \\ x_2x_1 & x_2x_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

- By construct, $\mathbf{Y}$ is positive semidefinite (PSD): $\mathbf{Y} \succeq \mathbf{0}$

$$\mathbf{v}^\top \mathbf{Y} \mathbf{v} = \mathbf{v}^\top \mathbf{x}\mathbf{x}^\top \mathbf{v} = (\mathbf{v}^\top \mathbf{x})^2 \geq 0.$$

- Using $\mathbf{Y}$ is PSD and rank-1

$$\max_{\mathbf{Y}} -\langle \mathbf{A}, \mathbf{Y} \rangle \text{ s.t. } \mathbf{Y} \succeq \mathbf{0}, \ \text{rank}\,(\mathbf{Y}) = 1, \ Y_{ii} = 1 \ \forall i.$$

SDP relaxation

- Original MAXCUT

$$\max_{\mathbf{Y}} -\langle \mathbf{A}, \mathbf{Y} \rangle \ \text{s.t.} \ \mathbf{Y} \succeq \mathbf{0}, \ \text{rank}\,(\mathbf{Y}) = 1, \ Y_{ii} = 1 \ \forall i.$$

- Relaxing the rank-1 constraint

$$\max_{\mathbf{Y}} -\langle \mathbf{A}, \mathbf{Y} \rangle \ \text{s.t.} \ \mathbf{Y} \succeq \mathbf{0}, \ Y_{ii} = 1 \ \forall i.$$

- Let $\mathbf{C} = -\mathbf{A}$

$$\max_{\mathbf{Y}} \langle \mathbf{C}, \mathbf{Y} \rangle \ \text{s.t.} \ \mathbf{Y} \succeq \mathbf{0}, \ Y_{ii} = 1 \ \forall i.$$

- The Relaxed problem is a SDP problem.
- Fact: for the cost value c, we have $c_{\text{original}} \leq c_{\text{relax}}$

Short summary

- MAXCUT in words: given undirected graph with no self-loop, find the cut that gives the largest capacity.
- MAXCUT expressed using adjacency matrix:

$$\max_{\mathbf{x}} \frac{1}{2} \sum_{i,j} A_{ij}(1 - x_i x_j) \text{ s.t. } x_i \in \{-1, +1\} \ \forall i.$$

or

$$\max_{\mathbf{x}} \frac{1}{4} \sum_{i,j} A_{ij}(x_i - x_j)^2 \text{ s.t. } x_i \in \{-1, +1\} \ \forall i.$$

- MAXCUT as quadratic problem on hypercube

$$\max_{\mathbf{x}} \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \text{ s.t. } x_i \in \{-1, +1\} \ \forall i.$$

- SDP relaxation of MAXCUT

$$\max_{\mathbf{Y}} \langle \mathbf{C}, \mathbf{Y} \rangle \text{ s.t. } \mathbf{Y} \succeq \mathbf{0}, \ Y_{ii} = 1 \ \forall i.$$

LP vs SDP

- LP($\mathbf{c}, \mathbf{a}_i, b_i$)

$$\min_{\mathbf{x}} \langle \mathbf{c}, \mathbf{x} \rangle \ \text{ s.t. } \ \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i, \ \mathbf{x} \geq \mathbf{0}.$$

  - $\mathbf{x} \in \mathbb{R}^n$ is a vector variable
  - $\mathbf{c} \in \mathbb{R}^n$
  - $\langle \mathbf{c}, \mathbf{x} \rangle = \mathbf{c}^\top \mathbf{x} = \sum c_i x_i$
  - $\mathbf{a}_i \in \mathbb{R}^n, i = 1, 2, \ldots, m$
  - $\mathbf{x} \geq \mathbf{0}$ means $\mathbf{x}$ is nonnegative vector

- SDP($\mathbf{C}, \mathbf{A}_i, b_i$)

$$\min_{\mathbf{X}} \langle \mathbf{C}, \mathbf{X} \rangle \ \text{ s.t. } \ \langle \mathbf{A}_i, \mathbf{X} \rangle = b_i, \ \mathbf{X} \succeq \mathbf{0}.$$

  - $\mathbf{X} \in \mathbb{R}^{n \times n}$ is a matrix variable
  - $\mathbf{C} \in \mathbb{R}^{n \times n}$
  - $\langle \mathbf{C}, \mathbf{X} \rangle = \operatorname{Tr} \mathbf{C}^\top \mathbf{X} = \sum C_{ij} X_{ij}$
  - $\mathbf{A}_i \in \mathbb{R}^{n \times n}, i = 1, 2, \ldots, m$
  - $\mathbf{X} \succeq \mathbf{0}$ means $\mathbf{x}$ is positive semidefinite

Inner product of matrix

- Linear Algebra fact: $\langle \mathbf{C}, \mathbf{X} \rangle = \sum_{i,j} C_{ij} X_{ij} = \mathrm{Tr}\,(\mathbf{C}^\top \mathbf{X})$.

  Inner product $\langle \mathbf{C}, \mathbf{X} \rangle$ is the "sum of elementwise product"

$$\left\langle \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \right\rangle = \sum_{ij} \left(\begin{bmatrix} c_{11}x_{11} & c_{12}x_{12} \\ c_{21}x_{21} & c_{22}x_{22} \end{bmatrix} \right)$$

$$\sum_{i,j} C_{ij} X_{ij} = \sum_{i} [\mathbf{C}^\top \mathbf{X}]_{ii} = \mathrm{Tr}\,(\mathbf{C}^\top \mathbf{X})$$

$$\underbrace{\begin{bmatrix} c_{11} & c_{21} \\ c_{12} & c_{22} \end{bmatrix}}_{\mathbf{C}^\top} \underbrace{\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}}_{\mathbf{X}} = \begin{bmatrix} c_{11}x_{11} + c_{21}x_{21} & * \\ * & c_{12}x_{12} + c_{22}x_{22} \end{bmatrix}$$

SDP is the generalization of LP

- SDP is similar to LP in many aspects except $\mathbf{x} \geq \mathbf{0}$ becomes $\mathbf{X} \succeq \mathbf{0}$.
- Definition (Positive semidefinite matrix)
  A matrix $\mathbf{M}$ is PSD if the quadratic form $\mathbf{x}^\top \mathbf{M}\mathbf{x} \geq 0$ for all $\mathbf{x}$.
- A diagonal matrix $\mathbf{D}$ is PSD if its diagonal is nonnegative.
- SDP

$$\min_{\mathbf{X}} \langle \mathbf{C}, \mathbf{X} \rangle \ \text{ s.t. } \ \langle \mathbf{A}_i, \mathbf{X} \rangle = b_i, \ \mathbf{X} \succeq \mathbf{0}.$$

  If $\mathbf{X}, \mathbf{C}, \mathbf{A}$ are diagonal matrix, SDP reduces to LP.

How to solve SDP

- General purpose solver: Interior point method (which was first proposed to solve LP!).
- Write your own code: first-order method.
- Both are not the focus in this course.
- *In this course you will not be tested on solving SDP problems.

End of document