Chance-Constrained Maximum Clique Problem

Shunyu Yao

Department of Business Analytics, University of Iowa

Joint work with Yongzheng Dai, Neng Fan and Pavlo Krokhmal

Mixed Integer Programming Workshop 2026

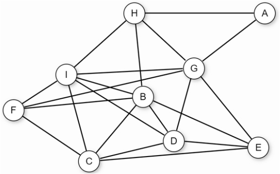
May 18, 2026

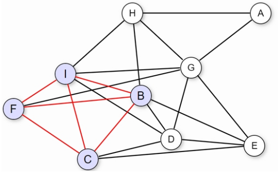
Maximum Clique Problem (MCP)

Clique: a subset of vertices in G where every distinct vertex pair is connected by an edge.

Problem Statement

Given a simple graph G, find a maximum clique (a clique with the largest possible number of vertices) of G.



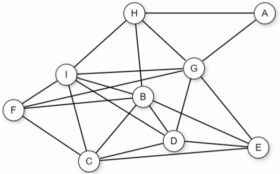


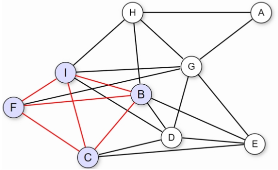
- A maximum clique is {B,C,F,I} with $\omega(G) = 4$, not unique.
- NP-complete, fixed-parameter intractable, hard to approx. within $n^{1-\epsilon}$ if NP $\neq$ ZPP.
- Many applications: social networks, bioinformatics, and computational chemistry, etc.

The cardinality of a maximum clique is called the clique number of G, denoted by $\omega(G)$.

Introduce Uncertainty in Clique Problems

MCP: Given a simple graph G, find a maximum clique of G.

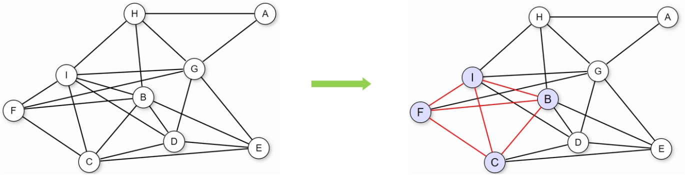




- (**Motivation**) Connections can be inherently uncertain in real-world networks.
- Thus, our goal is to identify a subset that is more likely to induce a larger clique.

Introduce Uncertainty in Clique Problems

MCP: Given a simple graph G, find a maximum clique of G.



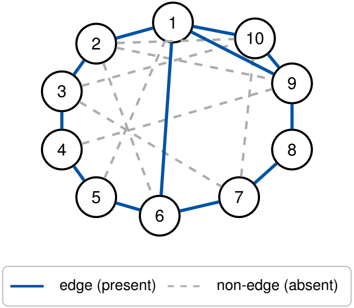
- (**Motivation**) Connections can be inherently uncertain in real-world networks.
- Thus, our goal is to identify a subset that is more likely to induce a larger clique.

- binary (adjacent/non-adjacent) relationship $\rightarrow$ binary random variable
- $e \in E(G) \rightarrow \mathbb{P}(e$ appears in $G)$
- find a maximum set S in G such that the induced graph $G[S]$ is a clique $\rightarrow$ find a maximum set S such that $\mathbb{P}(G[S]$ is a clique in $G) \geq \theta$ (θ is a predefined parameter)

Random Graphs

The **random graph** $\mathcal{G}(n, (p_{ij}))$ is a probability space over graphs on $V = \{1, 2, \ldots, n\}$, where each potential edge (i, j) appears **independently** with probability p_{ij}.

A graph G in $\mathcal{G}(n, (p_{ij}))$



For every pair of vertices $1 \le i < j \le n$,

$$\mathbb{P}[(i, j) \in \mathcal{G}] = p_{ij}, \qquad 0 \le p_{ij} \le 1,$$

and these events are **mutually independent**.

Special case: Erdős–Rényi random graph $\mathcal{G}(n, p)$

When $p_{ij} = p$ for all $i < j$, each edge appears **independently** with the same probability p.

Probability of a graph in $\mathcal{G}(n, p)$

For a deterministic graph $G = (V(G), E(G))$,

$$\mathbb{P}[\mathcal{G} = G] = p^{|E(G)|}(1 - p)^{\binom{n}{2} - |E(G)|}.$$

Chance-Constrained Maximum Clique Problem (CC-MCP)

Formally, we can define a **Chance-constrained maximum clique problem (CC-MCP)** as follows:

$$\text{[CC-MCP]} \quad \omega_\theta(\mathcal{G}) = \max_{S \subseteq V} \{|S| : \mathbb{P}[S \text{ is a clique in } \mathcal{G}] \geq \theta\}, \tag{1}$$

where $\theta \in [0, 1]$ is a prespecified threshold defined by the user.

- When $\theta = 0$, the problem is trivial and the optimal solution $S = V$.
- When $\theta = 1$, the problem becomes the classical MCP on graph $G^1 = (V, E^1)$, where the edge set $E^1 = \{e \in K_n : p_e = 1\}$.

Chance-Constrained Maximum Clique Problem (CC-MCP)

Formally, we can define a **Chance-constrained maximum clique problem (CC-MCP)** as follows:

$$\text{[CC-MCP]} \;\; \omega_\theta(\mathcal{G}) = \max_{S \subseteq V} \{|S| : \; \mathbb{P}\left[S \text{ is a clique in } \mathcal{G}\right] \geq \theta\}, \tag{1}$$

where $\theta \in [0, 1]$ is a prespecified threshold defined by the user.

Equivalent form:

$$\text{[CC-MCP]} \;\; \omega_\theta(\mathcal{G}) = \max \sum_{i \in V} x_i \tag{2a}$$

$$\text{s.t. } \mathbb{P}[x_i + x_j \leq 1 \; \forall (i,j) \in \bar{\mathcal{G}}] \geq \theta \tag{2b}$$

$$x_i \in \{0, 1\}, \; \forall i \in V \tag{2c}$$

where $\bar{\mathcal{G}}(n, (q_{ij}))$ denotes the complement of $\mathcal{G}(n, (p_{ij}))$ with $\mathbb{P}[(i,j) \in \bar{\mathcal{G}}] = q_{ij}$, $q_{ij} = 1 - p_{ij}, 1 \leq i < j \leq n$.

Mixed-Integer Programming (MIP) Reformulation

Theorem 1

CC-MCP admits the following exact MIP reformulation:

$$\max \sum_{i \in V} x_i \tag{3a}$$

$$\text{s.t.} \sum_{e \in K_n} y_e \log(1/p_e) \leq \log(1/\theta), \tag{3b}$$

$$y_e \leq x_i, y_e \leq x_j \ \forall e = (i,j) \in K_n \tag{3c}$$

$$x_i + x_j - 1 \leq y_e, \ \forall e = (i,j) \in K_n \tag{3d}$$

$$x_i, x_j \in \{0,1\}, 0 \leq y_e \leq 1, \ \forall e = (i,j) \in K_n \tag{3e}$$

- Constraints (3c) – (3e): linearization of the product $y_e := x_i x_j$, where $e = (i,j) \in K_n$.

Mixed-Integer Programming (MIP) Reformulation (cont.)

Theorem 1

CC-MCP admits the following exact MIP reformulation:

$$\max \sum_{i \in V} x_i \tag{3a}$$

$$\text{s.t.} \sum_{e \in K_n} y_e \log(1/p_e) \leq \log(1/\theta), \tag{3b}$$

$$y_e \leq x_i, y_e \leq x_j \ \forall e = (i,j) \in K_n \tag{3c}$$

$$x_i + x_j - 1 \leq y_e, \ \forall e = (i,j) \in K_n \tag{3d}$$

$$x_i, x_j \in \{0,1\}, 0 \leq y_e \leq 1, \ \forall e = (i,j) \in K_n \tag{3e}$$

Observations

- Note that constraint (3b) is a fractional knapsack problem (FKP) constraint.
  $\rightarrow y_e$ is a continuous variable in constraint (3b).

Mixed-Integer Programming (MIP) Reformulation (cont.)

Theorem 1

CC-MCP admits the following exact MIP reformulation:

$$\max \sum_{i \in V} x_i \tag{3a}$$

$$\text{s.t.} \sum_{e \in K_n} y_e \log(1/p_e) \le \log(1/\theta), \tag{3b}$$

$$y_e \le x_i, y_e \le x_j \ \forall e = (i,j) \in K_n \tag{3c}$$

$$x_i + x_j - 1 \le y_e, \ \forall e = (i,j) \in K_n \tag{3d}$$

$$x_i, x_j \in \{0,1\}, 0 \le y_e \le 1, \ \forall e = (i,j) \in K_n \tag{3e}$$

Observations

- Note that constraint (3b) is a fractional knapsack problem (FKP) constraint.
- If constraints (3c) & (3d) are removed, the resulting MIP can be solved in a very effective way.

Lagrangian Relaxation

Proposition 1

Constraints (3c) *can be removed from the MIP formulation* (3).

Dualize on the constraints (3d) and consider the Lagrangian relaxation (LR) problem:

$$Z(\lambda) := \max \quad \sum_{i \in V} \sum_{j \in N(i)} x_i \left(\frac{1}{|N(i)|} - \lambda_{ij} \right) + \sum_{e \in K_n} \lambda_e (y_e + 1) \tag{4a}$$

$$\text{s.t.} \quad \sum_{e \in K_n} y_e \log\left(\frac{1}{p_e}\right) \le \log\left(\frac{1}{\theta}\right) \tag{4b}$$

$$x \in \{0,1\}^n, 0 \le y_e \le 1, \forall e \in K_n \tag{4c}$$

Proposition 2

For a fixed $\bar{\lambda}$, $Z(\bar{\lambda})$ *can be computed in* $O(n^2)$ *time.*

Lagrangian Dual (LD) Problem

An optimal value of LR problem (4) is an upper bound for $\omega_\theta(\mathcal{G})$. To find the tightest upper bound over the Lagrangian multiplier vector λ, we consider the following Lagrangian dual (LD) problem:

$$Z_{LD} = \min_{\lambda \geq 0} Z(\lambda)$$

We can apply a standard subgradient algorithm to determine a near-optimal multiplier vector λ, i.e., compute the new multipliers by

$$\lambda^{k+1} := \text{Proj}_{\mathbb{R}^m_+}(\lambda^k - \gamma_k s^k), \tag{5}$$

- s^k is a subgradient of $Z(\lambda)$ at point λ^k
- $\text{Proj}_{\mathbb{R}^m_+}(\cdot)$ is the Euclidean projection on $\mathbb{R}^m_+$
- γ_k is a step length

Lagrangian cuts

Observation

For any multiplier $\tilde{\lambda}$, an optimal value of LR provides a valid upper bound for maximization problems, i.e., $\omega_\theta(\mathcal{G}) \leq Z(\tilde{\lambda})$.

Proposition 3

For any fixed multiplier $\tilde{\lambda}$, the Lagrangian cut

$$\sum_{i \in V} \left(\sum_{j:(i,j) \in E} \tilde{\lambda}_{ij} \right) x_i \leq \sum_{(i,j) \in E} \tilde{\lambda}_{ij} + \underbrace{\max_{\substack{0 \leq y \leq 1, \\ \sum w_{ij} y_{ij} \leq w'}} \sum_{(i,j)} \tilde{\lambda}_{ij} y_{ij}}_{\text{FKP problem}} . \tag{6}$$

is valid for the problem (3).

Delayed Constraint Generation (DCG)

We start by solving the following relaxed problem:

$$\max\left\{\sum_{i\in V} x_i : \text{(3b)}, \text{(3e)}, x_i + x_j - 1 \le y_e,\ \forall t, e = (i,j) \in E^t \subseteq K_n\right\}, \tag{7}$$

where E^t represents the edge set that violates constraints (3d) during the tth iteration.

- (Feasibility Check) if each edge $e \in K_n$ satisfies the CC-MCP constraint (3d), we have arrived at an optimal solution.
- (Cut Generation) o.w., add the Lagrangian cut (6) and constraint $x_i + x_j - 1 \le y_e$ to the relaxed problem and repeat.

The DCG algorithm will converge to an optimal solution in at most $O(n^2)$ iterations due to $\binom{n}{2}$ constraints (3d).

Experiment Settings

- Networks:
  - Complete graphs $n \in \{50, 100, 150, 200\}$
  - Real-world networks
- Parameter settings:
  - Threshold $\theta_i \in \{0.3, 0.5, 0.7, 0.9\}$
  - Half deterministic edges; remaining: $p_{ij} \sim$ Uniform(0.8,1)
  - Time limit - 1h
- Methods
  - DP (dynamic programming, Miao et al., 2014[1])
  - Default (solved directly by Gurobi)
  - DCG (w/o LCs)
  - DCG+LC (DCG with LCs)

[1] Miao, Zhuqi, Balabhaskar Balasundaram, and Eduardo L. Pasiliao. ”An exact algorithm for the maximum probabilistic clique problem.” Journal of Combinatorial Optimization 28.1 (2014): 105-120.

Numerical Results for Complete Graphs

Parameters		DP		Default			DCG			DCG+LC		
n	θ	Val	Time	Val	Time	Gap(%)	Val	Time	Gap(%)	Val	Time	Gap(%)
50	0.3	13	309.78	13	17.30	0.00	**13**	**1.40**	**0.00**	13	41.66	0.00
50	0.5	12	16.63	12	8.38	0.00	**12**	**0.95**	**0.00**	12	19.25	0.00
50	0.7	11	1.60	11	3.88	0.00	**11**	**0.64**	**0.00**	11	0.64	0.00
50	0.9	**9**	**0.04**	9	1.61	0.00	9	0.66	0.00	9	0.85	0.00
100	0.3	14	>1h	14	>1h	71.43	15	>1h	26.67	**16**	**458.78**	**0.00**
100	0.5	14	>1h	14	>1h	42.86	**14**	**766.08**	**0.00**	14	1875.38	0.00
100	0.7	**13**	**111.03**	13	>1h	15.38	13	657.10	0.00	13	893.55	0.00
100	0.9	**11**	**2.23**	11	101.42	0.00	11	27.89	0.00	11	35.84	0.00
150	0.3	**16**	>1h	14	>1h	185.71	15	>1h	193.33	**16**	>1h	**50.00**
150	0.5	15	>1h	13	>1h	169.23	15	3580.43	0.00	**15**	**1418.37**	**0.00**
150	0.7	13	>1h	12	>1h	158.33	14	1473.15	0.00	**14**	**1295.66**	**0.00**
150	0.9	**12**	**32.52**	12	1268.06	0.00	12	189.98	0.00	12	207.06	0.00
200	0.3	14	>1h	12	>1h	450.00	**17**	>1h	**182.35**	16	>1h	343.75
200	0.5	14	>1h	13	>1h	276.92	14	>1h	335.71	**15**	>1h	280.00
200	0.7	**14**	>1h	13	>1h	253.85	13	>1h	353.85	**14**	>1h	335.71
200	0.9	**13**	**141.27**	12	>1h	100.00	13	953.61	0.00	13	1215.24	0.00

Numerical Results for Real-World Networks

Parameters		DP		Default			DCG			DCG+LC		
Graph	θ	Val	Time	Val	Time	Gap(%)	Val	Time	Gap(%)	Val	Time	Gap(%)
C125-9	0.3	14	$>$ 1h	14	$>$ 1h	50.00	15	1615.34	0.00	**15**	**871.92**	**0.00**
C125-9	0.5	13	3390.33	13	$>$ 1h	30.77	13	483.12	0.00	**13**	**319.29**	**0.00**
C125-9	0.7	**12**	**144.95**	12	1970.35	0.00	12	253.14	0.00	12	177.98	0.00
C125-9	0.9	**11**	**2.59**	11	252.39	0.00	11	54.15	0.00	11	51.99	0.00
c-fat200-5	0.3	15	$>$ 1h	15	1974.23	0.00	15	414.03	0.00	**15**	**390.95**	**0.00**
c-fat200-5	0.5	13	695.91	13	890.77	0.00	13	298.68	0.00	**13**	**281.87**	**0.00**
c-fat200-5	0.7	**12**	**38.12**	12	868.06	0.00	12	234.63	0.00	12	211.34	0.00
c-fat200-5	0.9	**10**	**1.11**	10	1129.48	0.00	10	59.93	0.00	10	41.72	0.00
gen200-p0-9-55	0.3	14	$>$ 1h	15	$>$ 1h	153.33	15	$>$ 1h	66.67	**15**	$>$ 1h	**53.33**
gen200-p0-9-55	0.5	14	$>$ 1h	13	$>$ 1h	161.54	15	2677.23	0.00	**15**	**2167.80**	**0.00**
gen200-p0-9-55	0.7	14	2704.27	12	$>$ 1h	158.33	**14**	**922.37**	**0.00**	14	961.33	0.00
gen200-p0-9-55	0.9	**12**	**43.81**	11	$>$ 1h	81.82	12	366.06	0.00	12	381.22	0.00
san200-0-9-1	0.3	14	$>$ 1h	14	$>$ 1h	178.57	**16**	$>$ 1h	**50.00**	**16**	$>$ 1h	**50.00**
san200-0-9-1	0.5	13	$>$ 1h	13	$>$ 1h	176.92	15	3111.07	0.00	**15**	**2847.76**	**0.00**
san200-0-9-1	0.7	13	$>$ 1h	12	$>$ 1h	141.67	13	1116.00	0.00	**13**	**1037.27**	**0.00**
san200-0-9-1	0.9	**11**	**56.83**	10	$>$ 1h	110.00	11	663.42	0.00	11	381.16	0.00

Thanks for your attention!

Shunyu Yao
shunyu-yao@uiowa.edu