# A Branch-and-Cut Method for Solving the Bilevel Clique Interdiction Problem

Tim Becker

Rice University
Department of Computational and Applied Mathematics

November 1, 2015

# Outline

1. Introduction
   - Interdiction
   - Applications
2. Problem Formulation
   - Single Level
   - Bilevel
3. Algorithm
   - Finding the Next Clique
4. Numerical Results
5. Clique Relaxations
6. Conclusion

# Definitions

- **Graph**: $G = (V, E)$ where $V$ is the vertex set and $E$ is the edge set.
- **Clique**: A subset of the vertices such that each pair of vertices in the clique is connected by an edge.

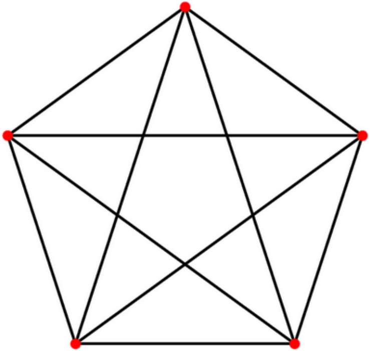


Figure: **Clique of size 5***

*Figure taken from `http://www.stephanspencer.com/how-graph-theory-relates/`.

# Interdiction

- **Interdiction**: A military term for removing the enemy's resources.
- Homeland Security:
  Associated cost is the amount of money/resources necessary to arrest a given person.
- Marketing:
  Associated cost is the amount of money/resources necessary to advertise to a given person.

# Bilevel Clique Interdiction Problem

- **Clique Interdiction Problem**: Maximize the number of cliques interdicted.
- **Bilevel Formulation**: Interdictor and Defender both attempt to optimize.
    - Defender minimizes the cliques that can be interdicted.

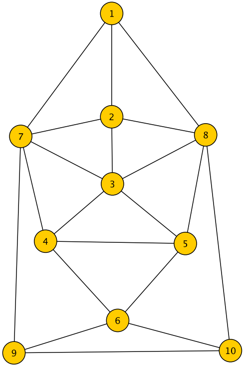
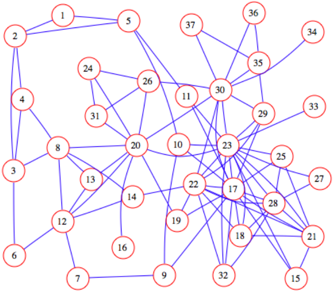



Figure: Sample Graph

# Terrorism Prevention

1) Alshehri
2) Sugami
3) Al-Marabh
4) Hijazi
5) W. Alshehri
6) A. Alghamdi
7) M. Alshehri
8) S. Alghamdi
9) Ahmed
10) Al-Hisawi
11) Al-Omari
12) H. Alghamdi
13) Alnami
14) Al-Haznawi
15) Darkazanli
16) Abdi
17) Al-Shehhi
18) Essabar
19) S. Alhazmi



20) N. Alhazmi
21) Bahaji
22) Jarrah
23) Atta
24) Shaikh
25) El Motassadeq
26) Al-Mihdhar
27) Moussaoui
28) Al-Shibh
29) Raissi
30) Hanjour
31) Awadallah
32) Budiman
33) Al-ani
34) Moqed
35) Abdullah
36) Al Salmi
37) Alhazmi

Figure: 9/11 Terrorist Network [1]

[1] Valdis E Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43–52, 2002

# Marketing

- Single Level Clique Interdiction:
  - Given a budget, how does one decide whom to market their product towards to maximize the spread of the advertisement?
- Examples of Bilevel Clique Interdiction:
  - No-Call List
  - Political Advertising

# Context

| | *Network* | *Clique* |
|---|---|---|
| *Single-Level* | ✓[2] | ✓[3] |
| *Bilevel* | ✓[4] | × |

Table: Interdiction Models

[2] Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002

[3] Foad Mahdavi Pajouh, Vladimir Boginski, and Eduardo L. Pasiliao. Maximum weighted clique network interdiction via vertex deletions. *Preprint submitted to Networks*, 2013

[4] R Kevin Wood. Bilevel network interdiction models: Formulations and solutions. *Wiley Encyclopedia of Operations Research and Management Science*, 2011

# Clique Interdiction Formulation

- $x_v = \begin{cases} 1 & \text{if node v is interdicted} \\ 0 & \text{otherwise.} \end{cases}$
- $y_k = \begin{cases} 1 & \text{if clique k is interdicted} \\ 0 & \text{otherwise.} \end{cases}$
- $I_k$ is the index set of all nodes within a given clique, $k$

- **Objective:**
  - Maximize $1^t y$
- **Constraints:**
  - $\sum_{v \in I_k} x_v \geq y_k, \quad \forall$ cliques k
  - $c^t x \leq R$
- **Bounds**:
  - $x_v \in \{0, 1\}, \quad \forall$ vertices v
  - $y_k \in \{0, 1\}, \quad \forall$ cliques k

# Bilevel Formulation

- $x_v = \begin{cases} 1 & \text{if node v is interdicted} \\ 0 & \text{otherwise.} \end{cases}$

- $y_k = \begin{cases} 1 & \text{if clique k is interdicted} \\ 0 & \text{otherwise.} \end{cases}$

- $z_v = \begin{cases} 1 & \text{if node v is defended} \\ 0 & \text{otherwise} \end{cases}$

- **Objective:**
  - Maximize $1^t y + 1^t z$
- **Constraints:**
  - $x \leq 1 - z$
  - $d^t z \leq Q$
  - $c^t x \leq R$
  - $\sum_{v \in I_k} x_v \geq y_k, \quad \forall$ cliques k
- **Bounds**:
  - $x_v \in \{0, 1\}, \quad \forall$ vertices v
  - $y_k \in \{0, 1\}, \quad \forall$ cliques k
  - $z_v \in \{0, 1\}, \quad \forall$ vertices v

# Linear Program Bilevel Formulation

- $x_v = \begin{cases} 1 & \text{if node v is interdicted} \\ 0 & \text{otherwise.} \end{cases}$
- $y_k = \begin{cases} 1 & \text{if clique k is interdicted} \\ 0 & \text{otherwise.} \end{cases}$
- $z_v = \begin{cases} 1 & \text{if node v is defended} \\ 0 & \text{otherwise.} \end{cases}$

- **Objective:**
  - Maximize $1^t y + 1^t z$
- **Constraints:**
  - $x \leq 1 - z$
  - $d^t z \leq Q$
  - $c^t x \leq R$
  - $\sum_{v \in I_k} x_v \geq y_k, \quad \forall$ cliques k
- **Bounds**:
  - $x_v \in [0, 1], \quad \forall$ vertices v
  - $y_k \in [0, 1], \quad \forall$ cliques k
  - $z_v \in [0, 1], \quad \forall$ vertices v

# LP Formulation Theorem

## Theorem

*At optimality, if $x_v$ is integral for all $v \in V$, then $y_k$ is integral for all cliques $k$.*

## Proof.

Assume not true for contradiction. Find a new feasible solution with greater objective value than optimal based on:

$$\sum_{v \in I_k} x_v \geq y_k, \quad \forall \text{ cliques k}$$

$\square$

# Algorithm Framework

Input: A graph file and budgets for interdictor and defender.

1. Remove all cliques from the formulation
2. Begin CLIQUE Function
    1. Solve linear relaxation in GUROBI
    2. Find a new clique to enter into the formulation
    3. Exit when no such new clique can be found
3. Branch on a node with fractional $x$ or $z$ value and return to CLIQUE
4. When no more cliques can be found and all values are integral, return

# Dual Formulation

Dual Variables:

- $\alpha : c'x \leq R$
- $\beta : d'x \leq Q$
- $p : x \leq 1 - z$
- $q : \sum_{v \in I_k} x_v \geq y_k$
- $s : x \leq 1$
- $t : y \leq 1$
- $u : z \leq 1$

- **Objective:**
  - Minimize $\alpha R + \beta Q + 1^t p + 1^t s + 1^t t + 1^t u$
- **Constraints:**
  - $s_v + \alpha c_v + p_v + \sum_{k | v \in I_k} q_k \geq 0$
  - $-q_k + t_k \geq 1$
  - $\beta d_v + p_v + u_v \geq 1$
- **Bounds**:
  - $\alpha, \beta, p_v, s_v, t_k, u_v \geq 0$
  - $q_k \leq 0$

# Choosing a Clique

- Choose cliques that break the second constraint: $-q_k + t_k \geq 1$
  - Consider only cliques, $k$, such that $q_k > -1$.

- Due to the removal of all cliques, $q_k$ does not exist until that clique has been introduced.

- Lower bound for $q_k$ from the first constraint: $s_v + \alpha c_v + p_v + \sum_{k|v \in I_k} q_k \geq 0$
  - $\min_{v \in I_k}\{-s_v - \alpha c_v - p_v - \sum_{k|v \in I_k} q_k\} \leq q_k$.

# Stopping Criterion

- Let $w_v = -s_v - \alpha c_v - p_v - \sum_{k|v \in I_k} q_k$
- Find a clique $k$ such that $-1 < min_{v \in I_k}\{w_v\}$
- Only choose nodes $v$ such that $-1 < w_v < 0$
- If no such nodes exist, STOP

# Algorithm Framework

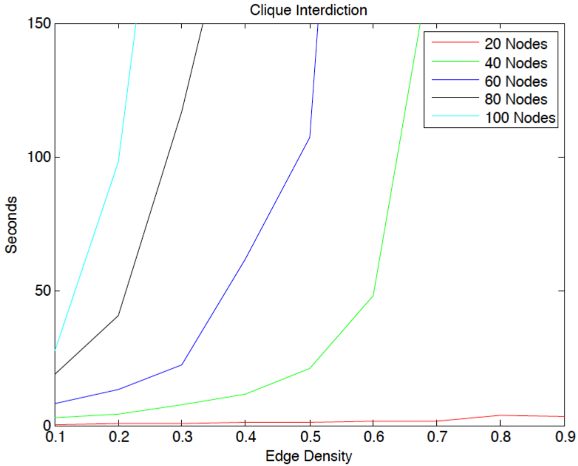
Input: A graph file and budgets for interdictor and defender.

1. Remove all cliques from the formulation
2. Begin CLIQUE Function
    1. Solve linear relaxation in GUROBI
    2. Find a new clique to enter into the formulation
    3. Exit when no such new clique can be found
3. Branch on a node with fractional $x$ or $z$ value and return to CLIQUE
4. When no more cliques can be found and all values are integral, return

# Computational Details

- Implemented in C++ and GUROBI
- Linux Desktop Computer with 3.10 GHz Intel Xeon Processor $E3 - 1220\ v2$
- All interdiction and defense costs were random rational numbers between 0 and $|V|$
- Each instance was run multiple times and averaged to find the approximate run time

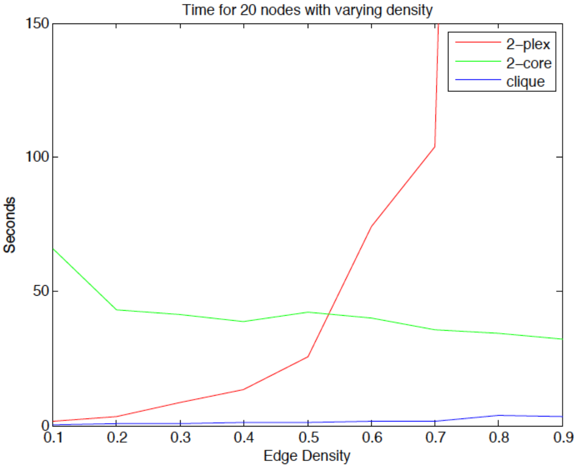
# Preliminary Numerical Results



# K-Plex & K-Core

- A subgraph $K \subseteq G$ is a $k$-plex if $\delta(G[K]) \geq |K| - k$
- A subgraph $K \subseteq G$ is a $k$-core if $|N(v) \cap V(K)| \geq k \;\; \forall v \in V(K)$.
- We look for maximal k-plexes and minimal k-cores.

# Preliminary Numerical Results



# Future Work

- Generalize Chvátal-Gomory cuts and determine valid inequalities.
- Introduce a defender minimizing cliques interdicted.
- Run further numerical simulations for K-Plex and K-Core.

# Summary

- Describe the Bilevel Clique Interdiction Problem and applications.
- Formulate the Bilevel Clique Interdiction Problem as an Integer Program and subsequent relaxed Linear Program.
- Develop an algorithm to solve the problem in C++, interfacing with GUROBI.
- Display numerical results for the algorithm.