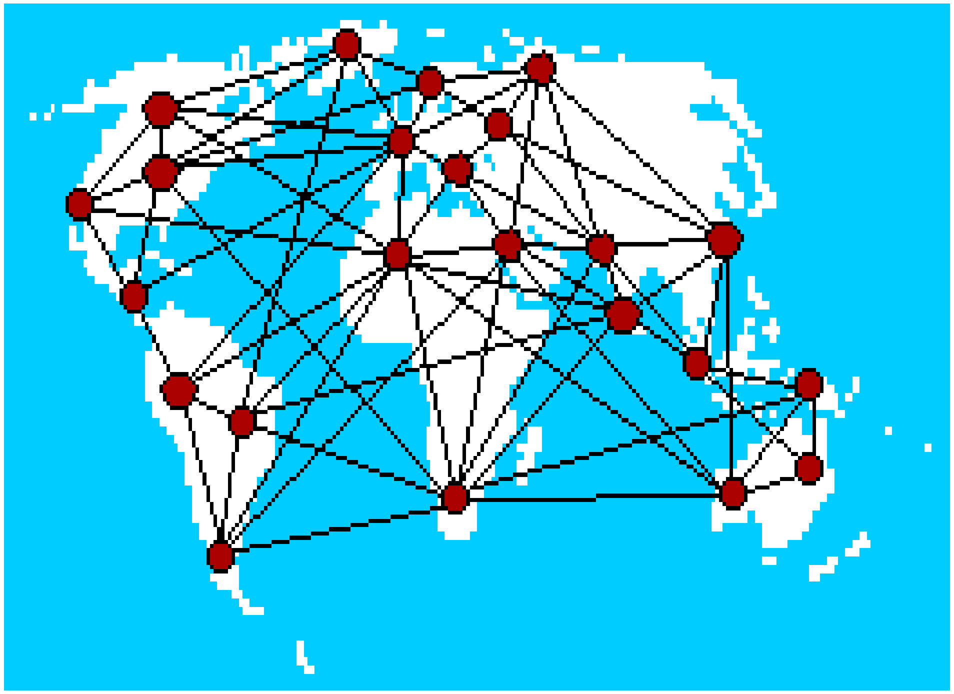
Critical Node Detection Problem

ITALY – May, 2008



Panos Pardalos
Distinguished Professor
CAO, Dept. of Industrial and Systems Engineering,
University of Florida

Outline of Talk

- Introduction
- Problem Definition
- Applications
- Proof NP Completeness
- Formulation
- Heuristics
- Results and Conclusions
- Future Direction

Introduction

- Acknowledgements
- Critical Node Detection
- Centrality
- Prestige
- Prominence
- Key Players

Introduction
Acknowledgments

- **Coauthors:** **A. Arulselvan, C. Commander, L. Elefteriadou**

Problem Definition

- Given a graph $G = (V,E)$ and an integer k
- Goal is to detect (delete) a set $|A| \leq k$ of critical nodes, or nodes whose deletion results in maximum pairwise disconnectivity
- Disconnectivity → MAX components subject to MIN difference in cardinality
- Example….

UNIVERSITY OF FLORIDA

Applications

- Assessing vulnerability of a supply chain network by determining the vital nodes
- First of many problems considered involving jamming/suppressing communication on a network
- Breakdown communication in covert networks
- Reduce transmissibility of virus and contagion of epidemic
- Drug design
- Emergency response

UNIVERSITY OF FLORIDA

Supply Chain Network

- It is essential to estimate the vulnerability of the supply chain network
- The study could be accommodated by maximizing the disconnectivity between supply and demand nodes instead of all pairs of nodes

Applications

- Assessing vulnerability of a supply chain network by determining the vital nodes
- First of many problems considered involving jamming/suppressing communication on a network
- Breakdown communication in covert networks
- Reduce transmissibility of virus and contagion of epidemic
- Drug design
- Emergency response

UNIVERSITY OF FLORIDA

Jamming Networks

- Given a graph whose arcs represent the communication links in the graph.
- (Offense) Select at most k nodes to target whose removal creates the maximum network disruption.
- (Defense) Determine which of your nodes to protect from enemy disruptions.
- Arulselvan, C., Pardalos, Shylo. Managing Network Risk Via Critical Node Detection. Risk Management in Telecommunication Networks, Gulpinar & Rustem (eds.), Springer, 2007.

UNIVERSITY OF FLORIDA

Applications

- **Assessing vulnerability of a supply chain network by determining the vital nodes**
- **First of many problems considered involving jamming/suppressing communication on a network**
- **Breakdown communication in covert networks**
- **Reduce transmissibility of virus and contagion of epidemic**
- **Drug design**
- **Emergency response**

UNIVERSITY OF FLORIDA

Covert/Terrorist Network

- **Use gathered intelligence to create social network interactions among terrorists**
- **Target those individuals whose "neutralization" will maximally disrupt the communication. (See example)**
- **Arulselvan, C., Elefteriadou, Pardalos. Detecting Critical Nodes in Sparse Graphs, *Computers and Operations Research*, 2008.**

UNIVERSITY OF FLORIDA

Applications

- Assessing vulnerability of a supply chain network by determining the vital nodes
- First of many problems considered involving jamming/suppressing communication on a network
- Breakdown communication in covert networks
- Reduce transmissibility of virus and contagion of epidemic
- Drug design
- Emergency response

UNIVERSITY OF FLORIDA

Controlling Social Contagion

- **Certain social populations have high rates of transmissibility of viruses.**
- **Mass vaccination is too expensive**
- **Determine the appropriate set of individuals to vaccinate so that the spread of the disease/virus is minimized**
- **Arulselvan, C., Elefteriadou, Pardalos. Detecting Critical Nodes in Sparse Graphs, *Computers and Operations Research*, 2008.**

UNIVERSITY OF FLORIDA

Applications

- Assessing vulnerability of a supply chain network by determining the vital nodes
- First of many problems considered involving jamming/suppressing communication on a network
- Breakdown communication in covert networks
- Reduce transmissibility of virus and contagion of epidemic
- Drug design
- Emergency response

Drug Design

- Examine protein-protein interaction maps.
- Determine which proteins to target in order to destroy the network.
- Last week, University of Florida researchers identify key protein interactions to target to destroy aggressive cancer cells' protective force field, University of Florida scientists reported this week at the American Association for Cancer Research's annual meeting in San Diego.

Applications

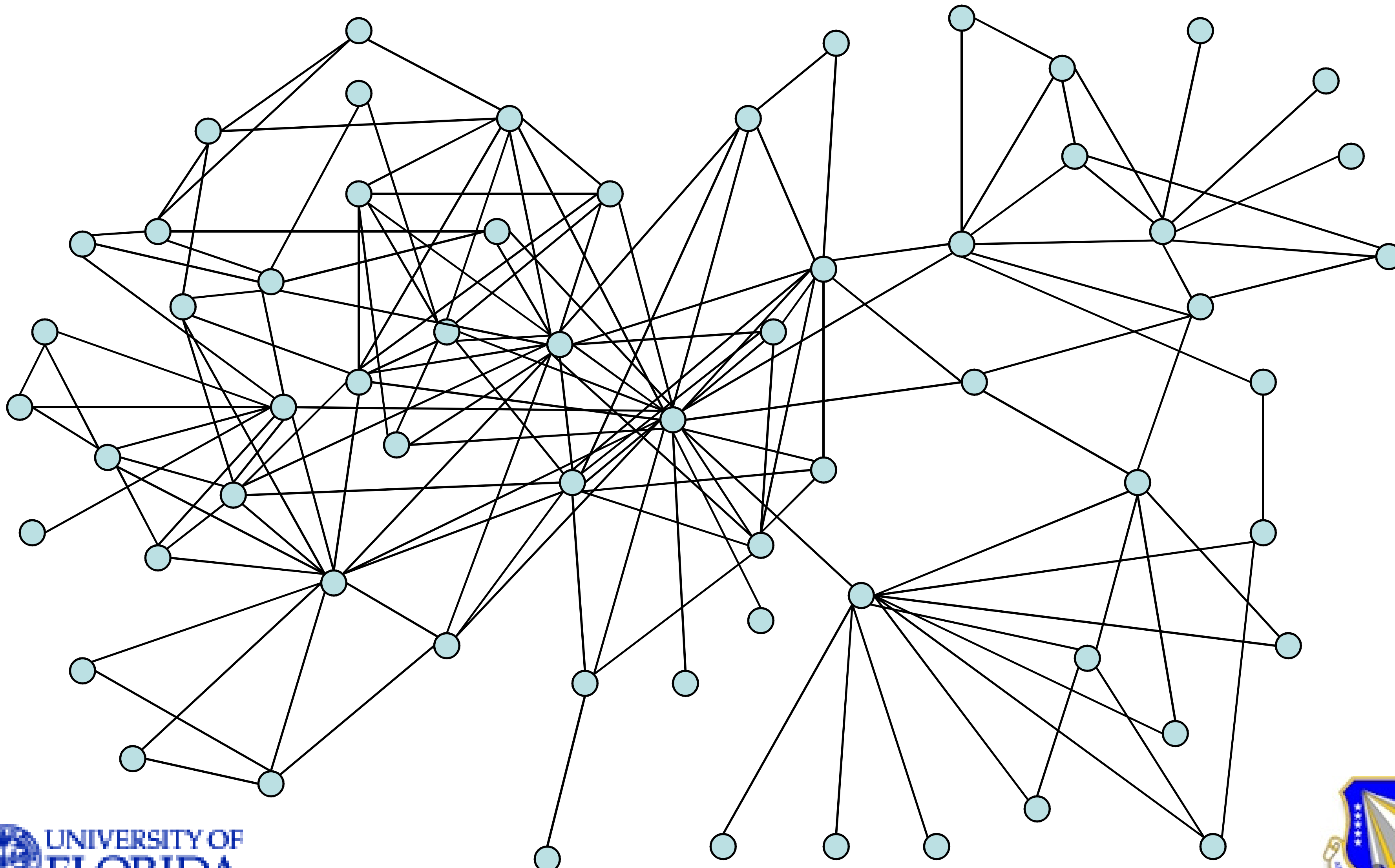
- Assessing vulnerability of a supply chain network by determining the vital nodes
- First of many problems considered involving jamming/suppressing communication on a network
- Breakdown communication in covert networks
- Reduce transmissibility of virus and contagion of epidemic
- Drug design
- Emergency response

UNIVERSITY OF FLORIDA

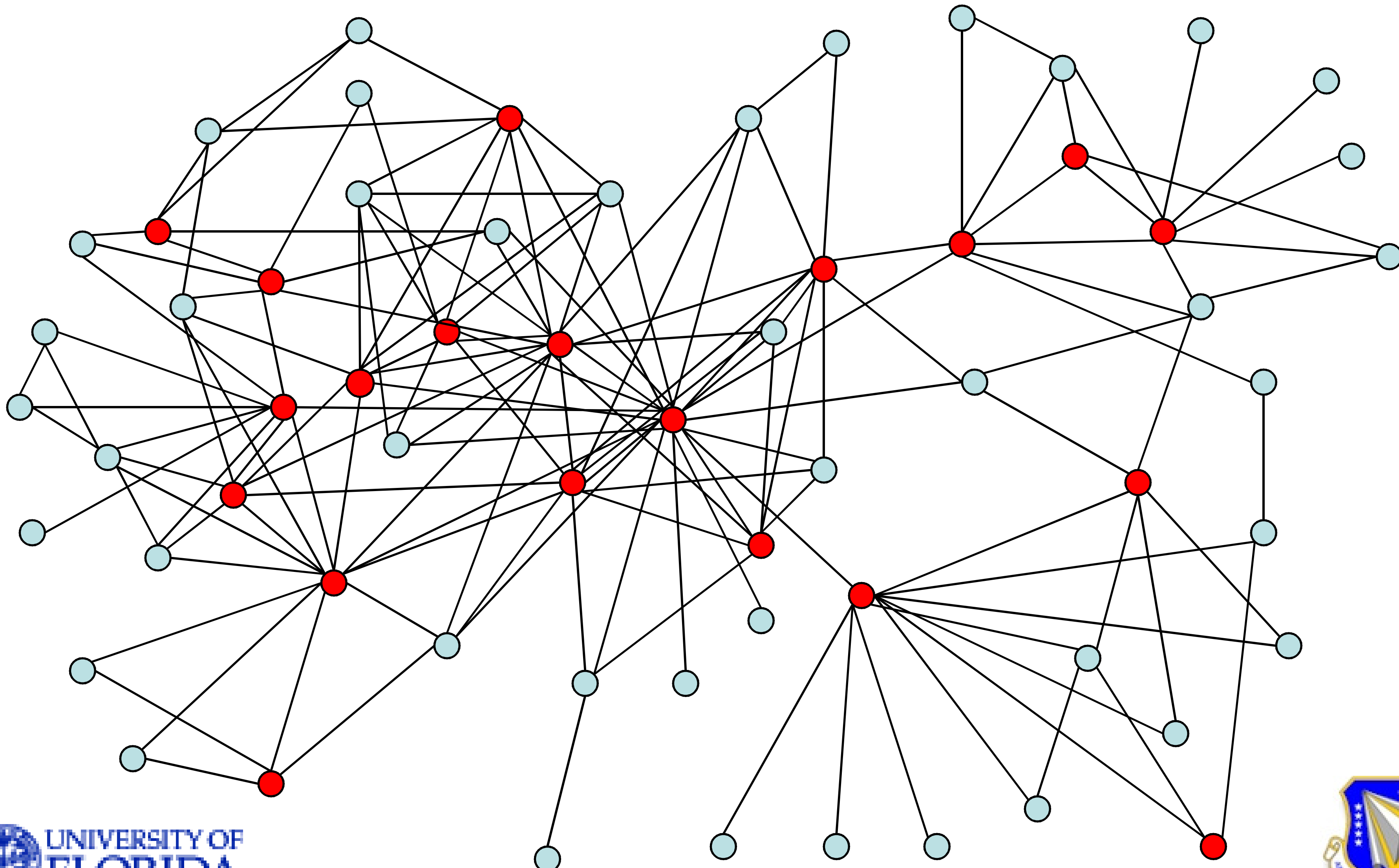
Emergency Response

- Identify roadways to attack to prevent enemy travel.
- Identify key roadways to fortify or repair first in the event of natural disaster.
- Enable mass evacuation (get out) and first responders (get in)
- RE: Hurricane Katrina

UNIVERSITY OF FLORIDA



UNIVERSITY OF FLORIDA



UNIVERSITY OF FLORIDA

Problem Definition

- **Decision Version: K-CNP**
- **Input: Undirected graph G = (V,E) and integer k**
- **Question: Is there a set *M*, where *M* is the set of all maximal connected components of G obtained by deleting *k* nodes or less, such that** $\sum_{\forall i \in M} \frac{\sigma_i(\sigma_i - 1)}{2} \leq K$

  where σ_i is the cardinality of component *i*, for all *i* in *M*?

Theoretical Results

- **Lemma 1:** **Let *M* be a partition of G = (V,E) in to *L* components obtained by deleting a set D, where |D| = k Then the objective function**

$$\sum_{\forall i \in M} \frac{\sigma_i(\sigma_i - 1)}{2} \geq \frac{(|V| - k)\left(\frac{|V| - k}{L} - 1\right)}{2}$$

with equality holding if and only $\sigma_i = \sigma_j$, for all i,j in M, where σ_i is the size of *i*th component of M.

- **Objective function is best when components are of average size.**

Theoretical Results

- **Lemma 2:** Let M_1 and M_2 be a two sets of partitions of $G = (V,E)$ obtained by deleting a set D1 and D2 sets of nodes respectively, where $|D_1| = |D_2| = k$. Let L_1 and L_2 be the number of components in M_1 and M_2 respectively, and $L_1 \geq L_2$. If $\sigma_i = \sigma_j$, for all i,j in M_1, then we obtain a better objective function value by deleting D_1.
- THE MORE (components), THE BETTER!!

Proof of NP-Completeness

- NP-complete: Reduction from Independent Set Problem by a simple transformation and the result follow from the above Lemmas.

Formulation

- Let $u_{i,j} = 1$, if i and j are in the same component of $G(V \setminus A)$, and 0 otherwise.
- Let $v_i = 1$, if node i is deleted in the optimal solution, and 0 otherwise.
- We can formulate the CNP as the following integer linear program

Formulation

$$
\begin{aligned}
\text{(CNP-1)} \quad \text{Minimize} \quad & \sum_{i,j \in V} u_{ij} \\
\text{s.t.} \quad & \\
& u_{ij} + v_i + v_j \geq 1, \ \forall \ (i,j) \in E, \\
& u_{ij} + u_{jk} - u_{ki} \leq 1, \ \forall \ (i,j,k) \in V, \\
& u_{ij} - u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V, \\
& -u_{ij} + u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V, \\
& \sum_{i \in V} v_i \leq k, \\
& u_{ij} \in \{0,1\}, \ \forall \ i,j \in V, \\
& v_i \in \{0,1\}, \ \forall \ i \in V.
\end{aligned}
$$

Formulation

$$
\text{(CNP-1)} \quad \text{Minimize} \quad \sum_{i,j \in V} u_{ij}
$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \ \forall \ (i,j) \in E,$$

$$u_{ij} + u_{jk} - u_{ki} \leq 1, \ \forall \ (i,j,k) \in V,$$

$$u_{ij} - u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V,$$

$$-u_{ij} + u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V,$$

$$\sum_{i \in V} v_i \leq k,$$

$$u_{ij} \in \{0,1\}, \ \forall \ i,j \in V,$$

$$v_i \in \{0,1\}, \ \forall \ i \in V.$$

If i and j in different components and there is an edge between them, at least one must be deleted

Formulation

$$\text{(CNP-1)} \quad \text{Minimize} \quad \sum_{i,j \in V} u_{ij}$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \ \forall\ (i,j) \in E,$$

$$u_{ij} + u_{jk} - u_{ki} \leq 1, \ \forall\ (i,j,k) \in V,$$

$$u_{ij} - u_{jk} + u_{ki} \leq 1, \ \forall\ (i,j,k) \in V,$$

$$-u_{ij} + u_{jk} + u_{ki} \leq 1, \ \forall\ (i,j,k) \in V,$$

$$\sum_{i \in V} v_i \leq k,$$

$$u_{ij} \in \{0,1\}, \ \forall\ i,j \in V,$$

$$v_i \in \{0,1\}, \ \forall\ i \in V.$$

Number of nodes deleted is at most k.

Formulation

$$(\text{CNP-1}) \quad \text{Minimize} \quad \sum_{i,j \in V} u_{ij}$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \ \forall \ (i,j) \in E,$$

$$u_{ij} + u_{jk} - u_{ki} \leq 1, \ \forall \ (i,j,k) \in V,$$

$$u_{ij} - u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V,$$

$$-u_{ij} + u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V,$$

$$\sum_{i \in V} v_i \leq k,$$

$$u_{ij} \in \{0,1\}, \ \forall \ i,j \in V,$$

$$v_i \in \{0,1\}, \ \forall \ i \in V.$$

For all triplets (i,j,k), if (i,j) in same comp and (j,k) in same comp, then (i,k) in same comp.

Formulation

$$
\begin{aligned}
\text{(CNP-1)} \quad \text{Minimize} \quad & \sum_{i,j \in V} u_{ij} \\
\text{s.t.} \quad & \\
& u_{ij} + v_i + v_j \geq 1, \ \forall \ (i,j) \in E, \\
& u_{ij} + u_{jk} - u_{ki} \leq 1, \ \forall \ (i,j,k) \in V, \\
& u_{ij} - u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V, \\
& -u_{ij} + u_{jk} + u_{ki} \leq 1, \ \forall \ (i,j,k) \in V, \\
& \sum_{i \in V} v_i \leq k, \\
& u_{ij} \in \{0,1\}, \ \forall \ i,j \in V, \\
& v_i \in \{0,1\}, \ \forall \ i \in V.
\end{aligned}
$$

Can combine into one constraint for a simpler model

Formulation

$$\text{(CNP-2)} \quad \text{Minimize} \quad \sum_{i,j \in V} u_{ij}$$

s.t.

$$u_{ij} + u_{jk} - u_{ki} \leq 1, \ \forall\ (i,j,k) \in V,$$

$$u_{ij} - u_{jk} + u_{ki} \leq 1, \ \forall\ (i,j,k) \in V,$$

$$-u_{ij} + u_{jk} + u_{ki} \leq 1, \ \forall\ (i,j,k) \in V,$$

$$u_{ij} + v_i + v_j \geq 1, \ \forall\ (i,j) \in E,$$

$$u_{ij} + u_{jk} + u_{ki} \neq 2, \ \forall\ (i,j,k) \in V,$$

$$\sum_{i \in V} v_i \leq k,$$

$$u_{ij} \in \{0,1\}, \ \forall\ i,j \in V,$$

$$v_i \in \{0,1\}, \ \forall\ i \in V,$$

Heuristics

- Notice if objective was only a function of the number of components then we could use approximation for Max K-Cut by modifying the Gomory-Hu tree
- BUT objective is also concerned with size of components
- Problem is harder…Too bad!

Heuristics

- Recall the objective function:
  - Minimize $\sum_{i,j} u_{ij}$, where $u_{ij} = 1$, if i and j in same component of vertex deleted subgraph.
- We can re-write this as follows:

$$\sum_{i \in S} \frac{s_i (s_i - 1)}{2}$$

- Where S is the set of all components and s_i is the size of the i-th component.
- Easily identify with DFS in $O(|V| + |E|)$ time!

Heuristics

- We implement a heuristic based on Maximal Independent Sets
  - Why? Because induced subgraph is empty
  - Maximum Independent Set provides upper bound on # of components in optimal solution.
- Greedy type procedure
- Enhanced with local search procedure
- Results are excellent
  - Heuristic obtains optimal solutions in fraction of time required by CPLEX
  - Runs in $O(k^2 + |V|k)$ time.

UNIVERSITY OF FLORIDA

Heuristics

procedure CriticalNode(G, k)

$\text{MIS} \leftarrow \texttt{MaximalIndepSet}(G)$

while $(|\text{MIS}| \neq |V| - k)$ **do**

$\quad i \leftarrow \arg\min\{\sum_{i \in S} \frac{s_i(s_i-1)}{2} : S \in G(\text{MIS} \cup \{i\}), i \in V \setminus \text{MIS}\}$

$\quad \text{MIS} \leftarrow \text{MIS} \cup \{i\}$

end while

return $V \setminus \text{MIS}$ /* set of k nodes to delete */

end procedure CriticalNode

1) **Find Maximal Independent Set (MIS)**
2) Repeat until we have found k critical nodes
3) Find node which returns best objective function value (GREEDY)
4) Add to MIS

UNIVERSITY OF FLORIDA

Heuristics

procedure $\texttt{CriticalNode}(G, k)$

$\quad \text{MIS} \leftarrow \texttt{MaximalIndepSet}(G)$

$\quad$ **while** $(|\text{MIS}| \neq |V| - k)$ **do**

$\quad\quad i \leftarrow \arg\min\{\sum_{i \in S} \frac{s_i(s_i-1)}{2} : S \in G(\text{MIS} \cup \{i\}), i \in V \setminus \text{MIS}\}$

$\quad\quad \text{MIS} \leftarrow \text{MIS} \cup \{i\}$

$\quad$ **end while**

$\quad$ **return** $V \setminus \text{MIS}$ $\quad$ /* set of k nodes to delete */

end procedure $\texttt{CriticalNode}$

1) **Find Maximal Independent Set (MIS)**
2) **Repeat until we have found k critical nodes**
3) Find node which returns best objective function value (GREEDY)
4) Add to MIS

UNIVERSITY OF FLORIDA

Heuristics

procedure $\texttt{CriticalNode}(G, k)$

$\quad \text{MIS} \leftarrow \texttt{MaximalIndepSet}(G)$

$\quad$ **while** $(|\text{MIS}| \neq |V| - k)$ **do**

$\quad\quad i \leftarrow \arg\min\{\sum_{i \in S} \frac{s_i(s_i-1)}{2} : S \in G(\text{MIS} \cup \{i\}), i \in V \setminus \text{MIS}\}$

$\quad\quad \text{MIS} \leftarrow \text{MIS} \cup \{i\}$

$\quad$ **end while**

$\quad$ **return** $V \setminus \text{MIS}$ $\quad$ /* set of k nodes to delete */

end procedure $\texttt{CriticalNode}$

1) **Find Maximal Independent Set (MIS)**
2) **Repeat until we have found k critical nodes**
3) **Find node which returns best objective function value (GREEDY)**
4) Add to MIS

Heuristics

procedure CriticalNode(G, k)

  $\text{MIS} \leftarrow$ MaximalIndepSet(G)

  while $(|\text{MIS}| \neq |V| - k)$ **do**

    $i \leftarrow \arg\min\{\sum_{i \in S} \frac{s_i(s_i - 1)}{2} : S \in G(\text{MIS} \cup \{i\}), i \in V \setminus \text{MIS}\}$

    $\text{MIS} \leftarrow \text{MIS} \cup \{i\}$

  end while

  return $V \setminus \text{MIS}$ /* set of k nodes to delete */

end procedure CriticalNode

1) **Find Maximal Independent Set (MIS)**
2) **Repeat until we have found k critical nodes**
3) **Find node which returns best objective function value (GREEDY)**
4) **Add to MIS**

Heuristics

```
procedure CriticalNodeLS(G, k)
    X* ← ∅
    f(X*) ← ∞
    for j = 1 to MaxIter do
        X ← CriticalNode(G, k)
        X ← LocalSearch(X)
        if f(X) < f(X*) then
            X* ← X
        end if
    end
    return (V \ X*)    /* set of k nodes to delete */
end procedure CriticalNodeLS
```

- Same procedure as before
- Insert local search
    - 2-exchange method
- Return best overall solution

UNIVERSITY OF FLORIDA

Heuristics

procedure CriticalNodeLS(G, k)
  $X^* \leftarrow \emptyset$
  $f(X^*) \leftarrow \infty$
  for $j = 1$ to MaxIter do
    $X \leftarrow$ CriticalNode(G, k)
    $X \leftarrow$ LocalSearch(X)
    if $f(X) < f(X^*)$ then
      $X^* \leftarrow X$
    end if
  end
  return $(V \setminus X^*)$ /* set of k nodes to delete */
end procedure CriticalNodeLS

- **Same procedure as before**
- **Insert local search**
  - **2-exchange method**
- **Return best overall solution**

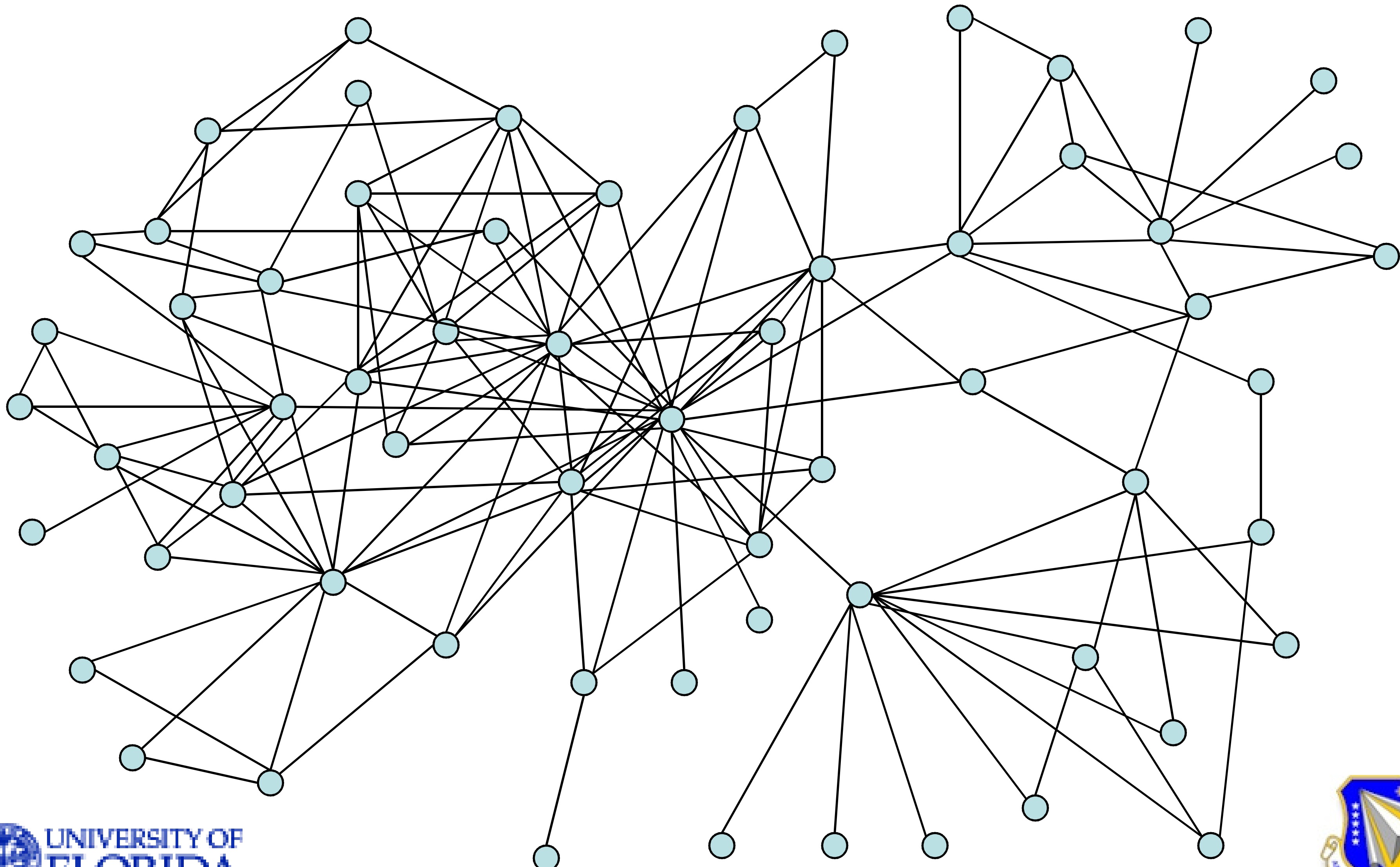
UNIVERSITY OF FLORIDA

Heuristics

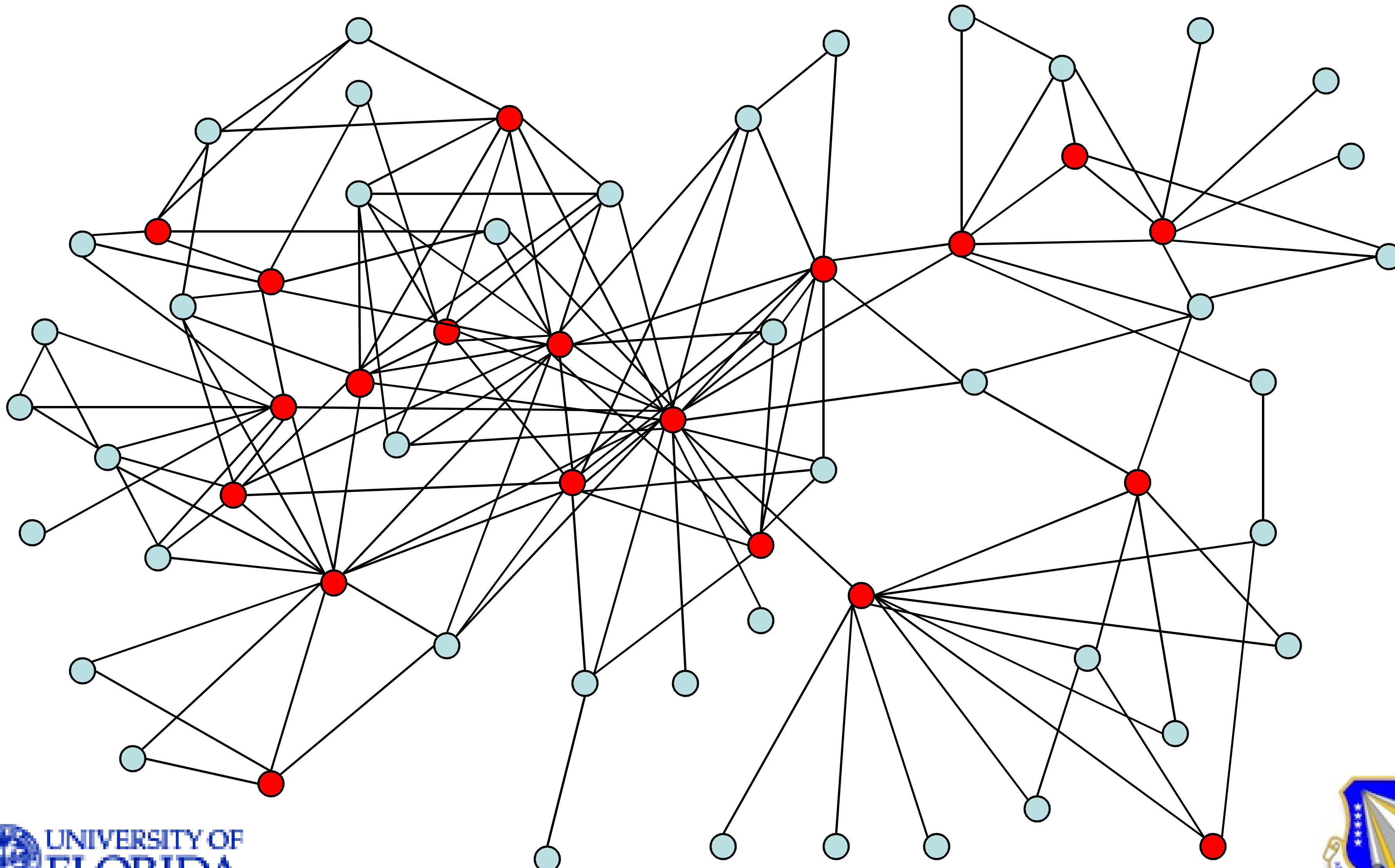
```
procedure CriticalNodeLS(G, k)
    X* ← ∅
    f(X*) ← ∞
    for j = 1 to MaxIter do
        X ← CriticalNode(G, k)
        X ← LocalSearch(X)
        if f(X) < f(X*) then
            X* ← X
        end if
    end
    return (V \ X*)    /* set of k nodes to delete */
end procedure CriticalNodeLS
```

- **Same procedure as before**
- **Insert local search**
    - **2-exchange method**
- **Return best overall solution**

Results



Results



Results

Instance	IP Model		Heuristic	
Nodes Deleted (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
20	20	12.69	20	0.01
15	61	277.77	61	0.01
10	169	3337.06	169	0.02
9	214	2792.33	214	0.02
8	282	15111.94	282	0.01
7	327	10792.08	327	0.01

- **This is the case you just saw!!**
- **Optimal solutions computed for all values of k for this terrorist graph**
- **The solutions are computed very quickly**
- **Wait…it gets better!**

UNIVERSITY OF FLORIDA

Results

Instance	IP Model		Heuristic	
Nodes Deleted (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
20	20	12.69	20	0.01
15	61	277.77	61	0.01
10	169	3337.06	169	0.02
9	214	2792.33	214	0.02
8	282	15111.94	282	0.01
7	327	10792.08	327	0.01

- **This is the case you just saw!!**
- **Optimal solutions computed for all values of k for this terrorist graph**
- **The solutions are computed very quickly**
- **Wait…it gets better!**

UNIVERSITY OF FLORIDA

Results

Instance	IP Model		Heuristic	
Nodes Deleted (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
20	20	12.69	20	0.01
15	61	277.77	61	0.01
10	169	3337.06	169	0.02
9	214	2792.33	214	0.02
8	282	15111.94	282	0.01
7	327	10792.08	327	0.01

- **This is the case you just saw!!**
- **Optimal solutions computed for all values of k for this terrorist graph**
- **The solutions are computed very quickly**
- **Wait…it gets better!**

Results

Instance	IP Model		Heuristic	
Nodes Deleted (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
20	20	12.69	20	0.01
15	61	277.77	61	0.01
10	169	3337.06	169	0.02
9	214	2792.33	214	0.02
8	282	15111.94	282	0.01
7	327	10792.08	327	0.01

- **This is the case you just saw!!**
- **Optimal solutions computed for all values of k for this terrorist graph**
- **The solutions are computed very quickly**
- **Wait…it gets better!**

Results

- **Random instances from 75-150 nodes**
- Various values of k for each set
- Optimal solution found for each instance
  - Average CPLEX time: 289.44 seconds
  - Average Heuristic time: 0.33 seconds

Instance			IP Model		Heuristic	
Nodes	Arcs	Deleted Nodes (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
75	140	20	36	66.7	36	0.03
75	140	25	18	33.28	18	0.03
75	140	30	7	4.23	7	0.04
75	210	25	26	93.71	26	0.04
75	210	30	8	3.57	8	0.05
75	210	35	2	4.36	2	0.04
75	280	33	26	749.19	26	0.04
75	280	35	20	164.34	20	0.06
75	280	37	13	83.98	13	0.11
100	194	25	44	151.14	44	0.09
100	194	30	20	59.66	20	0.11
100	194	35	10	8.51	10	0.12
100	285	40	23	136.47	23	0.11
100	285	42	17	263.82	17	0.17
100	285	45	11	16.78	11	0.23
100	380	45	22	128.13	22	0.15
100	380	47	16	243.07	16	0.16
100	380	50	10	228.72	10	0.11
125	240	33	62	5047.51	62	0.30
125	240	40	29	118.92	29	0.24
125	240	45	16	17.09	16	0.39
150	290	40	40	41.6	40	0.47
150	290	50	12	26.29	12	0.831
150	290	60	1	24.92	1	0.851
150	435	61	19	29.55	19	0.741
150	435	65	13	31.45	13	1.952
150	435	67	11	37.91	11	0.801

UNIVERSITY OF FLORIDA

Results

- Random instances from 75-150 nodes
- Various values of k for each set
- Optimal solution found for each instance
  - Average CPLEX time: 289.44 seconds
  - Average Heuristic time: 0.33 seconds

Instance			IP Model		Heuristic	
Nodes	Arcs	Deleted Nodes (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
75	140	20	36	66.7	36	0.03
75	140	25	18	33.28	18	0.03
75	140	30	7	4.23	7	0.04
75	210	25	26	93.71	26	0.04
75	210	30	8	3.57	8	0.05
75	210	35	2	4.36	2	0.04
75	280	33	26	749.19	26	0.04
75	280	35	20	164.34	20	0.06
75	280	37	13	83.98	13	0.11
100	194	25	44	151.14	44	0.09
100	194	30	20	59.66	20	0.11
100	194	35	10	8.51	10	0.12
100	285	40	23	136.47	23	0.11
100	285	42	17	263.82	17	0.17
100	285	45	11	16.78	11	0.23
100	380	45	22	128.13	22	0.15
100	380	47	16	243.07	16	0.16
100	380	50	10	228.72	10	0.11
125	240	33	62	5047.51	62	0.30
125	240	40	29	118.92	29	0.24
125	240	45	16	17.09	16	0.39
150	290	40	40	41.6	40	0.47
150	290	50	12	26.29	12	0.831
150	290	60	1	24.92	1	0.851
150	435	61	19	29.55	19	0.741
150	435	65	13	31.45	13	1.952
150	435	67	11	37.91	11	0.801

UNIVERSITY OF FLORIDA

Results

- Random instances from 75-150 nodes
- Various values of k for each set
- Optimal solution found for each instance
  - Average CPLEX time: 289.44 seconds
  - Average Heuristic time: 0.33 seconds

Instance			IP Model		Heuristic	
Nodes	Arcs	Deleted Nodes (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
75	140	20	36	66.7	36	0.03
75	140	25	18	33.28	18	0.03
75	140	30	7	4.23	7	0.04
75	210	25	26	93.71	26	0.04
75	210	30	8	3.57	8	0.05
75	210	35	2	4.36	2	0.04
75	280	33	26	749.19	26	0.04
75	280	35	20	164.34	20	0.06
75	280	37	13	83.98	13	0.11
100	194	25	44	151.14	44	0.09
100	194	30	20	59.66	20	0.11
100	194	35	10	8.51	10	0.12
100	285	40	23	136.47	23	0.11
100	285	42	17	263.82	17	0.17
100	285	45	11	16.78	11	0.23
100	380	45	22	128.13	22	0.15
100	380	47	16	243.07	16	0.16
100	380	50	10	228.72	10	0.11
125	240	33	62	5047.51	62	0.30
125	240	40	29	118.92	29	0.24
125	240	45	16	17.09	16	0.39
150	290	40	40	41.6	40	0.47
150	290	50	12	26.29	12	0.831
150	290	60	1	24.92	1	0.851
150	435	61	19	29.55	19	0.741
150	435	65	13	31.45	13	1.952
150	435	67	11	37.91	11	0.801

UNIVERSITY OF FLORIDA

Results

- Random instances from 75-150 nodes
- Various values of k for each set
- Optimal solution found for each instance
    - Average CPLEX time: 289.44 seconds
    - Average Heuristic time: 0.33 seconds

Instance			IP Model		Heuristic	
Nodes	Arcs	Deleted Nodes (k)	Objective Value	Execution Time (s)	Objective Value	Execution Time (s)
75	140	20	36	66.7	36	0.03
75	140	25	18	33.28	18	0.03
75	140	30	7	4.23	7	0.04
75	210	25	26	93.71	26	0.04
75	210	30	8	3.57	8	0.05
75	210	35	2	4.36	2	0.04
75	280	33	26	749.19	26	0.04
75	280	35	20	164.34	20	0.06
75	280	37	13	83.98	13	0.11
100	194	25	44	151.14	44	0.09
100	194	30	20	59.66	20	0.11
100	194	35	10	8.51	10	0.12
100	285	40	23	136.47	23	0.11
100	285	42	17	263.82	17	0.17
100	285	45	11	16.78	11	0.23
100	380	45	22	128.13	22	0.15
100	380	47	16	243.07	16	0.16
100	380	50	10	228.72	10	0.11
125	240	33	62	5047.51	62	0.30
125	240	40	29	118.92	29	0.24
125	240	45	16	17.09	16	0.39
150	290	40	40	41.6	40	0.47
150	290	50	12	26.29	12	0.831
150	290	60	1	24.92	1	0.851
150	435	61	19	29.55	19	0.741
150	435	65	13	31.45	13	1.952
150	435	67	11	37.91	11	0.801

UNIVERSITY OF FLORIDA

Cardinality Critical Node Problem (CCNP)

- **Alternate Formulation:**
  - Suppose now, we want to limit the connectivity of the agents.
  - We can impose a constraint for this.
  - Now, we minimize the number of nodes deleted to satisfy this constraint.
  - We have the **CARDINALITY CONSTRAINED CRITICAL NODE PROBLEM**

CCNP - Formulation

$$\text{Minimize} \quad \sum_{i \in V} v_i$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \ \forall \ (i,j) \in E,$$

$$u_{ij} + u_{jk} + u_{ki} \neq 2, \ \forall \ (i,j,k) \in V,$$

$$\sum_{i,j \in V} u_{ij} \leq L,$$

$$u_{ij} \in \{0,1\}, \ \forall \ i,j \in V,$$

$$v_i \in \{0,1\}, \ \forall \ i \in V,$$

CCNP - Formulation

Minimize deleted nodes

$$\text{Minimize} \quad \sum_{i \in V} v_i$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \; \forall \, (i,j) \in E,$$

$$u_{ij} + u_{jk} + u_{ki} \neq 2, \; \forall \, (i,j,k) \in V,$$

$$\sum_{i,j \in V} u_{ij} \leq L,$$

$$u_{ij} \in \{0,1\}, \; \forall \, i,j \in V,$$

$$v_i \in \{0,1\}, \; \forall \, i \in V,$$

CCNP - Formulation

$$\text{Minimize} \quad \sum_{i \in V} v_i$$

s.t.

$$u_{ij} + v_i + v_j \geq 1, \; \forall \, (i,j) \in E,$$

$$u_{ij} + u_{jk} + u_{ki} \neq 2, \; \forall \, (i,j,k) \in V,$$

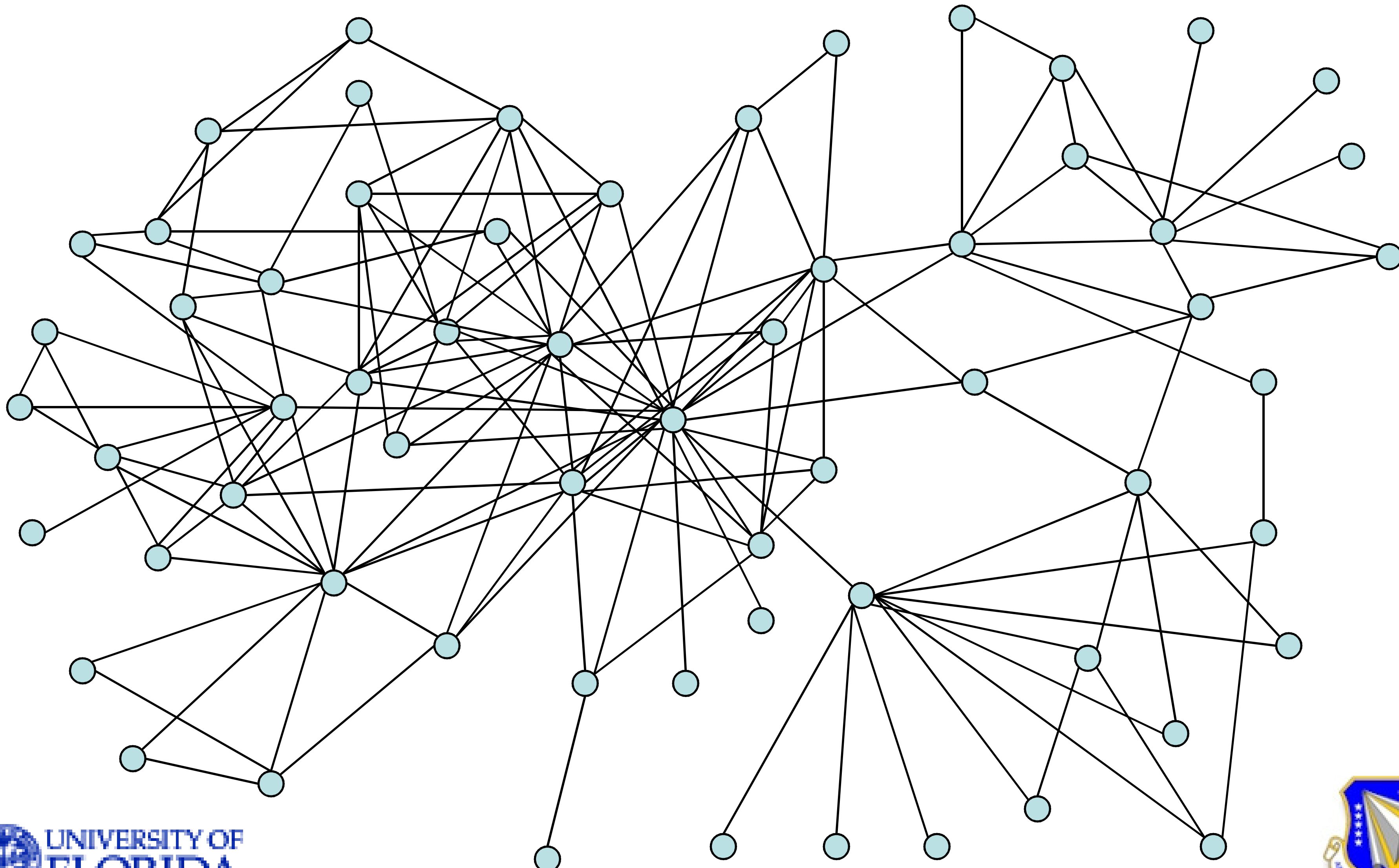
$$\sum_{i,j \in V} u_{ij} \leq L,$$

$$u_{ij} \in \{0,1\}, \; \forall \, i,j \in V,$$

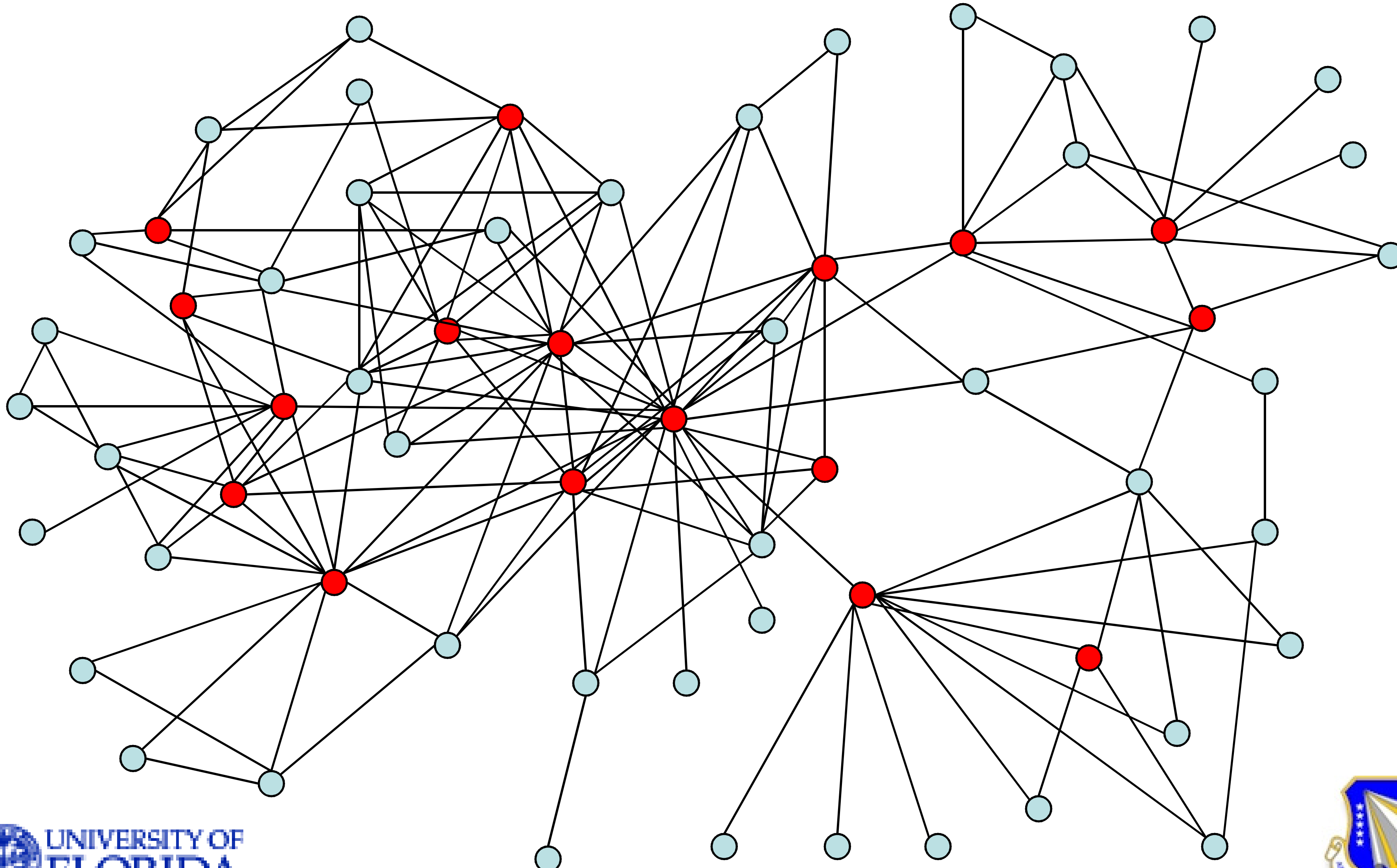
$$v_i \in \{0,1\}, \; \forall \, i \in V,$$

Connectivity constraint

CCNP - Example



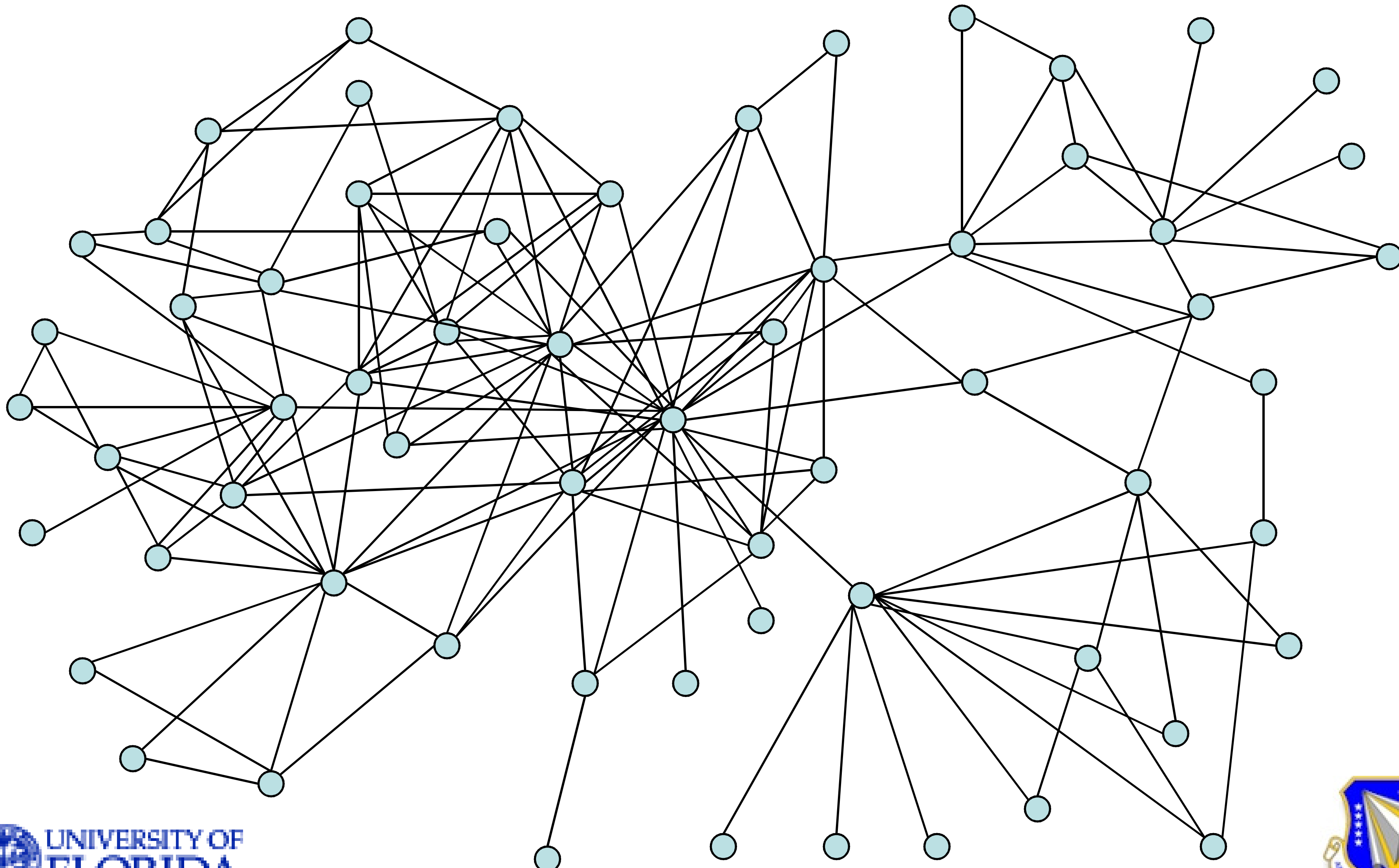
CCNP - Example



CCNP - Heuristics

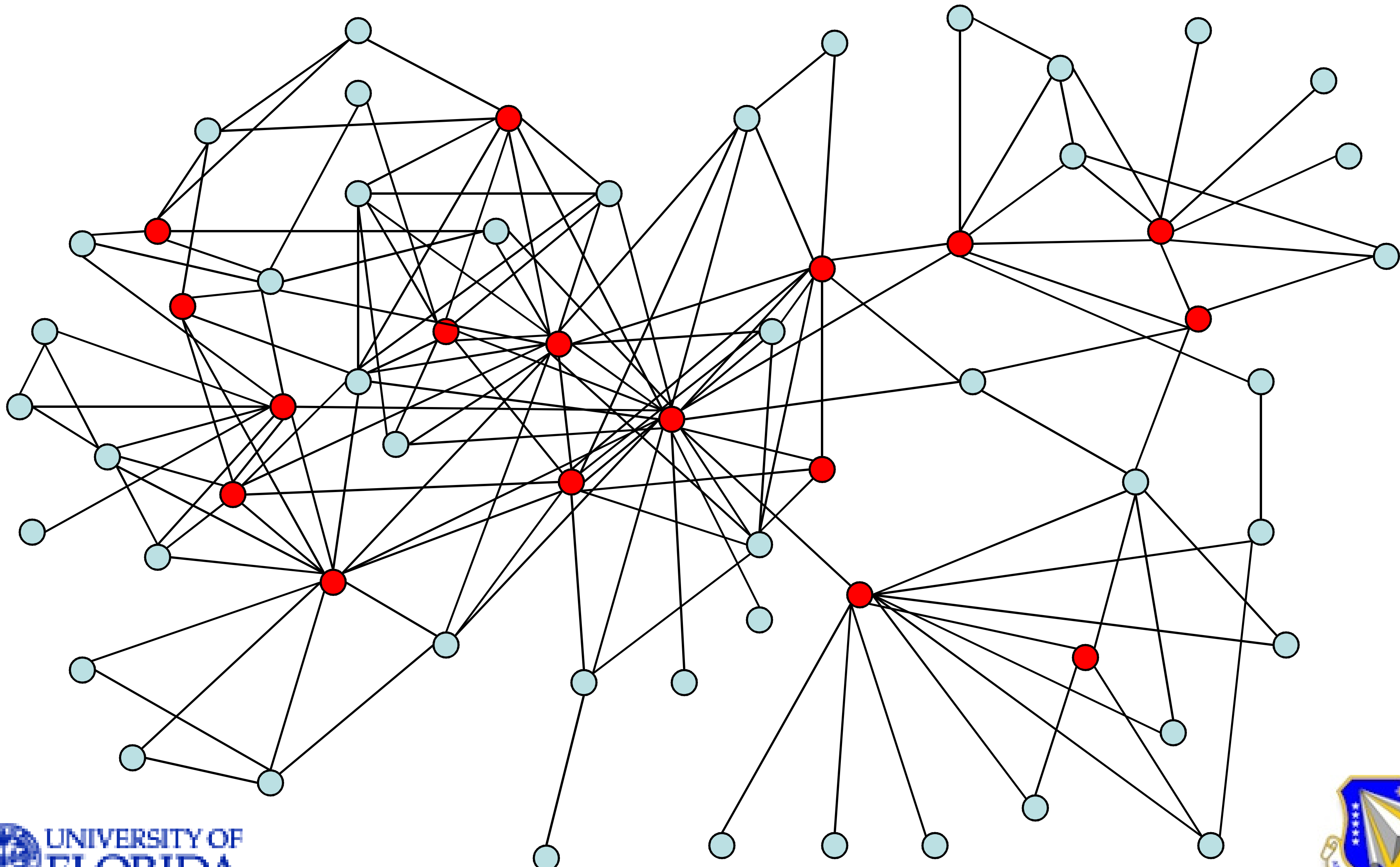
- We modified the MIS heuristic for this problem, but it is easy to create pathological instances. So…
- We also implemented a Genetic Algorithm for the CC-CNP.
- The GA was able to find optimal solutions for all instances tested.
- Example (again). Here L = 4. Opt Soln = 17.

Results



UNIVERSITY OF FLORIDA

Results



UNIVERSITY OF FLORIDA

Results

Instance	IP Model		Genetic Alg		ComAlg		ComAlg + LS	
Max Conn. Index (L)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)
3	21	188.98	21	0.25	22	0.01	21	0.1
4	17	886.09	17	0.741	19	0.01	17	0.45
5	15	30051.09	15	0.871	20	0.18	25	1.331
8	–	–	13	0.39	14	0.05	13	0.07
10	–	–	11	0.741	12	0.07	11	0.05

- This is the case you just saw!!
- Optimal solutions computed for all values of L for this terrorist graph
- The solutions are computed very quickly
- Wait…it gets better!

UNIVERSITY OF FLORIDA

Results

Instance	IP Model		Genetic Alg		ComAlg		ComAlg + LS	
Max Conn. Index (L)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)
3	21	188.98	21	0.25	22	0.01	21	0.1
4	17	886.09	17	0.741	19	0.01	17	0.45
5	15	30051.09	15	0.871	20	0.18	25	1.331
8	–	–	13	0.39	14	0.05	13	0.07
10	–	–	11	0.741	12	0.07	11	0.05

- **This is the case you just saw!!**
- **Optimal solutions computed for all values of L for this terrorist graph**
- **The solutions are computed very quickly**
- **Wait…it gets better!**

UNIVERSITY OF FLORIDA

Results

Instance	IP Model		Genetic Alg		ComAlg		ComAlg + LS	
Max Conn. Index (L)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)
3	21	188.98	21	0.25	22	0.01	21	0.1
4	17	886.09	17	0.741	19	0.01	17	0.45
5	15	30051.09	15	0.871	20	0.18	25	1.331
8	–	–	13	0.39	14	0.05	13	0.07
10	–	–	11	0.741	12	0.07	11	0.05

- This is the case you just saw!!
- Optimal solutions computed for all values of L for this terrorist graph
- The solutions are computed very quickly
- Wait…it gets better!

Results

Instance	IP Model		Genetic Alg		ComAlg		ComAlg + LS	
Max Conn. Index (L)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)	Obj Val	Comp Time (s)
3	21	188.98	21	0.25	22	0.01	21	0.1
4	17	886.09	17	0.741	19	0.01	17	0.45
5	15	30051.09	15	0.871	20	0.18	25	1.331
8	–	–	13	0.39	14	0.05	13	0.07
10	–	–	11	0.741	12	0.07	11	0.05

- **This is the case you just saw!!**
- **Optimal solutions computed for all values of L for this terrorist graph**
- **The solutions are computed very quickly**
- **Wait…it gets better!**

UNIVERSITY OF FLORIDA

Results

- Considered randomly generated instances with various values of L.
- Solution Quality:
  - GA: optimal solutions found for 100% of cases.
  - MIS Heuristic: optimal solutions found for 87.5% of cases.

Instance			IP Model		Genetic Alg		ComAlg + LS	
Nodes	Arcs	Max Conn. Index (L)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)
20	45	2	9	0.04	9	0.02	9	0.03
20	45	4	6	0.13	6	0.04	6	0.862
20	45	8	5	0.39	5	0.04	5	1.482
25	60	2	11	0.07	11	0.49	11	0.08
25	60	4	9	14.1	9	2.113	10	0.01
25	60	8	7	26.64	7	0.05	8	0.06
30	50	2	11	0.07	11	0.06	11	0.01
30	50	4	8	0.1	8	0.05	8	0
30	50	8	6	1152.15	6	0.09	6	0
30	75	4	10	18.77	10	0.14	10	0.02
30	75	6	9	442.41	9	0.09	9	0.04
30	75	10	7	64.94	7	0.18	8	0
35	60	2	12	0.13	12	0.14	12	0.14
35	60	4	8	29.89	8	0.711	8	0
35	60	6	7	31.61	7	0.31	7	0.01
40	70	2	15	0.17	15	0.1	15	0.101
40	70	4	11	341.97	11	0.06	11	0
40	70	6	8	78.94	8	0.2	8	0.04
45	80	2	16	0.24	16	0.06	16	0.1
45	80	4	11	48.17	11	0.05	11	0.02
45	80	6	8	118.23	8	0.09	8	0.071
50	135	2	19	0.36	19	0.27	19	0.05
50	135	4	15	165.18	15	0.63	15	0.291
50	135	6	14	5722.88	14	0.721	14	0.03
Total (Sum)			24	8257.58	24	6.705	27	3.417

UNIVERSITY OF FLORIDA

Results

- **Considered randomly generated instances with various values of L.**
- **Solution Quality:**
  - **GA: optimal solutions found for 100% of cases.**
  - MIS Heuristic: optimal solutions found for 87.5% of cases.

Instance			IP Model		Genetic Alg		ComAlg + LS	
Nodes	Arcs	Max Conn. Index (L)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)
20	45	2	9	0.04	9	0.02	9	0.03
20	45	4	6	0.13	6	0.04	6	0.862
20	45	8	5	0.39	5	0.04	5	1.482
25	60	2	11	0.07	11	0.49	11	0.08
25	60	4	9	14.1	9	2.113	10	0.01
25	60	8	7	26.64	7	0.05	8	0.06
30	50	2	11	0.07	11	0.06	11	0.01
30	50	4	8	0.1	8	0.05	8	0
30	50	8	6	1152.15	6	0.09	6	0
30	75	4	10	18.77	10	0.14	10	0.02
30	75	6	9	442.41	9	0.09	9	0.04
30	75	10	7	64.94	7	0.18	8	0
35	60	2	12	0.13	12	0.14	12	0.14
35	60	4	8	29.89	8	0.711	8	0
35	60	6	7	31.61	7	0.31	7	0.01
40	70	2	15	0.17	15	0.1	15	0.101
40	70	4	11	341.97	11	0.06	11	0
40	70	6	8	78.94	8	0.2	8	0.04
45	80	2	16	0.24	16	0.06	16	0.1
45	80	4	11	48.17	11	0.05	11	0.02
45	80	6	8	118.23	8	0.09	8	0.071
50	135	2	19	0.36	19	0.27	19	0.05
50	135	4	15	165.18	15	0.63	15	0.291
50	135	6	14	5722.88	14	0.721	14	0.03
Total (Sum)			24	8257.58	24	6.705	27	3.417

UNIVERSITY OF FLORIDA

Results

- **Considered randomly generated instances with various values of L.**
- **Solution Quality:**
  - GA: optimal solutions found for 100% of cases.
  - MIS Heuristic: optimal solutions found for 87.5% of cases.

Instance			IP Model		Genetic Alg		ComAlg + LS	
Nodes	Arcs	Max Conn. Index (L)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)
20	45	2	9	0.04	9	0.02	9	0.03
20	45	4	6	0.13	6	0.04	6	0.862
20	45	8	5	0.39	5	0.04	5	1.482
25	60	2	11	0.07	11	0.49	11	0.08
25	60	4	9	14.1	9	2.113	10	0.01
25	60	8	7	26.64	7	0.05	8	0.06
30	50	2	11	0.07	11	0.06	11	0.01
30	50	4	8	0.1	8	0.05	8	0
30	50	8	6	1152.15	6	0.09	6	0
30	75	4	10	18.77	10	0.14	10	0.02
30	75	6	9	442.41	9	0.09	9	0.04
30	75	10	7	64.94	7	0.18	8	0
35	60	2	12	0.13	12	0.14	12	0.14
35	60	4	8	29.89	8	0.711	8	0
35	60	6	7	31.61	7	0.31	7	0.01
40	70	2	15	0.17	15	0.1	15	0.101
40	70	4	11	341.97	11	0.06	11	0
40	70	6	8	78.94	8	0.2	8	0.04
45	80	2	16	0.24	16	0.06	16	0.1
45	80	4	11	48.17	11	0.05	11	0.02
45	80	6	8	118.23	8	0.09	8	0.071
50	135	2	19	0.36	19	0.27	19	0.05
50	135	4	15	165.18	15	0.63	15	0.291
50	135	6	14	5722.88	14	0.721	14	0.03
Total (Sum)			24	8257.58	24	6.705	27	3.417

UNIVERSITY OF FLORIDA

Results

- **Contributions of the chapter**
- **K-CNP**
  - Propose math program based on integer linear programming.
  - Proof of computational complexity
  - Implement an efficient heuristic based on maximal independent sets
  - Heuristic finds optimal solutions for all instances tested in fraction of time required by CPLEX
- **CC-CNP**
  - Math Programming formulation
  - Genetic Algorithm implemented finds optimal solutions for all instances tested.
- **Current Work**
  - Weighted version of the problem
  - Approximation of the problem
- **Papers:**
  - A. Arulselvan, C.W. Commander, L. Elefteriadou, P.M. Pardalos. Detecting critical nodes in social networks. *Computers and Operations Research*, 2008.
  - A. Arulselvan, C.W. Commander, P.M. Pardalos, O. Shylo. Managing network risk via critical node identification. *Risk Management in Telecommunication Networks*, N. Gulpinar and B. Rustem (editors), Springer, to appear 2008 (in process)

UNIVERSITY OF FLORIDA

Conclusions and Future Directions

- Identified nodes of sparse
- Breakdown communication
- Integer Programming and Heuristics
- Approximation algorithms
- Weighted version of the problems

UNIVERSITY OF FLORIDA

THANK YOU!!!!!

QUESTIONS?