

Recent advances in critical element detection problems

Ashwin Arulselvan

**Department of Management Science
University of Strathclyde**

Workshop on Clustering and Search techniques for large scale networks

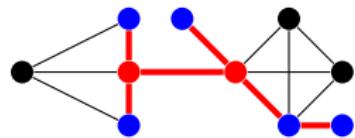
Oct 25, 2015

Outline

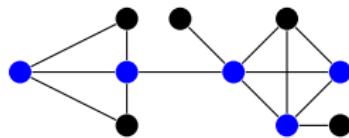
- ▶ Various definitions in the literature
- ▶ Three definitions of recent interest
- ▶ What has happened so far?
- ▶ Where are we going?

Traditional graph theory problems

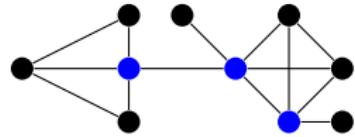
Steiner Tree



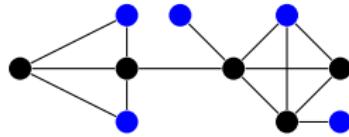
Vertex cover



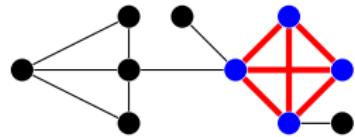
Dominating sets



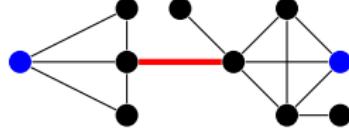
Independent Sets



Cliques



Cuts



Generalisation of these ideas

- ▶ Multicuts and multiway cuts
- ▶ Feedback vertex/edge cover
- ▶ Centrality
- ▶ Prestige
- ▶ Prominence
- ▶ k-plex, quasi-cliques
- ▶ clique cover, path cover

Generalisation of these ideas

- ▶ Nodes (edges) on deletion that causes fragmentation
- ▶ Set of nodes (edges) on inclusion induces a dense subgraph

One such generalisation

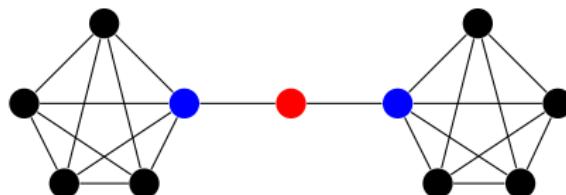
- ▶ pairwise connectivity of the remaining nodes (deletion - ISP)
- ▶ size of the largest connected component (deletion - VC)
- ▶ densest induced subgraph (inclusion - Clique)

Applications

- ▶ Jamming/suppressing communication on a network
- ▶ Reduce transmissibility of virus and contagion of epidemic
- ▶ Drug design
- ▶ Emergency response - shelter location
- ▶ Coalitions in social networks
- ▶ Influential individuals in social networks

Problem Definition

- ▶ Decision Version: $K - CNP$
- ▶ Input: Undirected graph $G = (V, E)$ and integers k
- ▶ Output: Find a subset $A \subset V$ of k nodes, whose deletion results in maximum pairwise connectivity



Objective function motivation

Lemma

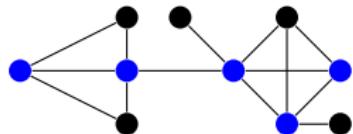
Let M be a partition of $G = (V, E)$ into L components obtained by deleting a set D of nodes, where $|D| = k$. Then the objective function $\sum_{h \in M} \frac{(\sigma_h(\sigma_h - 1))}{2} \geq \frac{(|V| - k)^2}{2(L-1)}$, with equality holding if and only if $\sigma_h = \sigma_I, \forall h, I \in M$, where σ_h is the size of h^{th} component of M .

Lemma

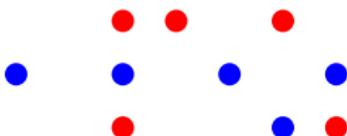
Let M_1 and M_2 be two sets of partitions obtained by deleting D_1 and D_2 sets of nodes, respectively, from graph $G = (V, E)$, where $|D_1| = |D_2| = k$. Let L_1 and L_2 be the number components in M_1 and M_2 , respectively, and $L_1 \geq L_2$. If $\sigma_h = \sigma_I, \forall h, I \in M_1$, then we obtain a better objective function value by deleting the set D_1 .

NP-Completeness

k -vertex cover



$(|V| - k)$ -Independent set



A reduction from k -VCP (Vertex cover problem) or $(|V| - k)$ -ISP (independent set problem)

Approximation hardness (rough idea)

An algorithm is α -approximate if it terminates with a solution with a value $\leq \alpha OPT$.

A is a NP-hard decision problem and $A \leq_P B$ such that

Yes: With value at most a

No: With value at least b

We cannot have $\frac{b}{a}$ approximation algorithm for problem B

Approximation hopes for CNP

- ▶ ISP (YES instance) 0
- ▶ ISP (NO instance) ≥ 1

UNLIKELY! (Planar graphs, degree bounded graphs)

Positive result

Polynomially solvable in bounded treewidths with unit weights

Di Summa et al. (2013)

Fixed parameter tractability

- ▶ Given a parameter k as an input and size of the input is n
- ▶ Is there an algorithm to an NP-complete problem that runs in $f(k)O(n^c)$, where $f(k)$ is a function (possibly exponential) in k .

What about CNP?

Negative on this end too! Through a reduction from clique cover problem

Hermelin et. al. (2015)

Formulation

min

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

s.t.

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

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

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

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

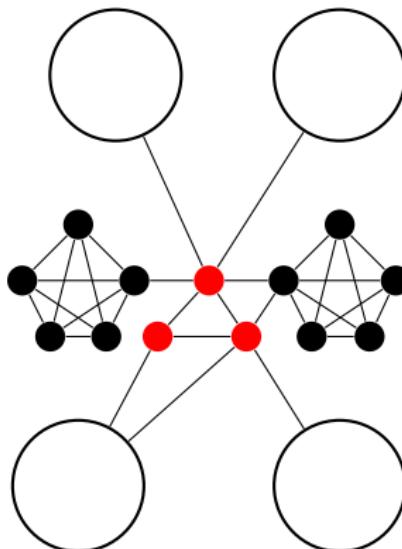
$$\sum_{i \in V} v_i \leq k,$$

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

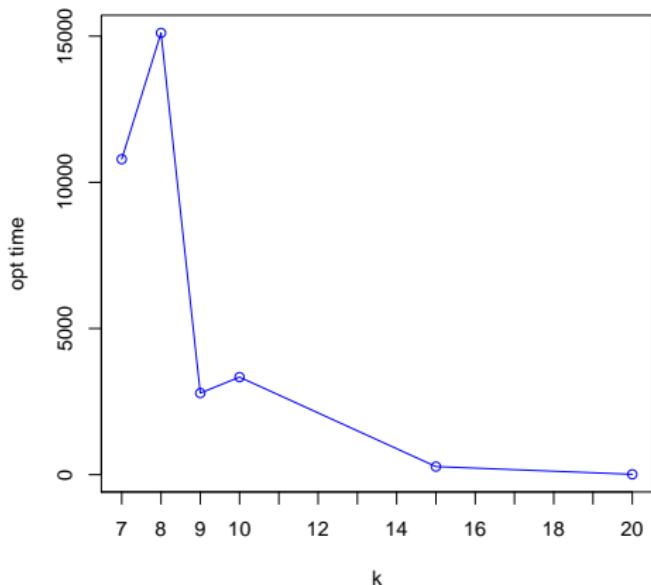
$$v_i \in \{0, 1\}, \quad \forall i \in V.$$

Heuristics

- ▶ Find a maximal independent set (MIS)
- ▶ Add as many vertices as possible “greedily” to the independent set
- ▶ Improve it with Local search
- ▶ Repeat the trials with different starting MIS and pick the best one

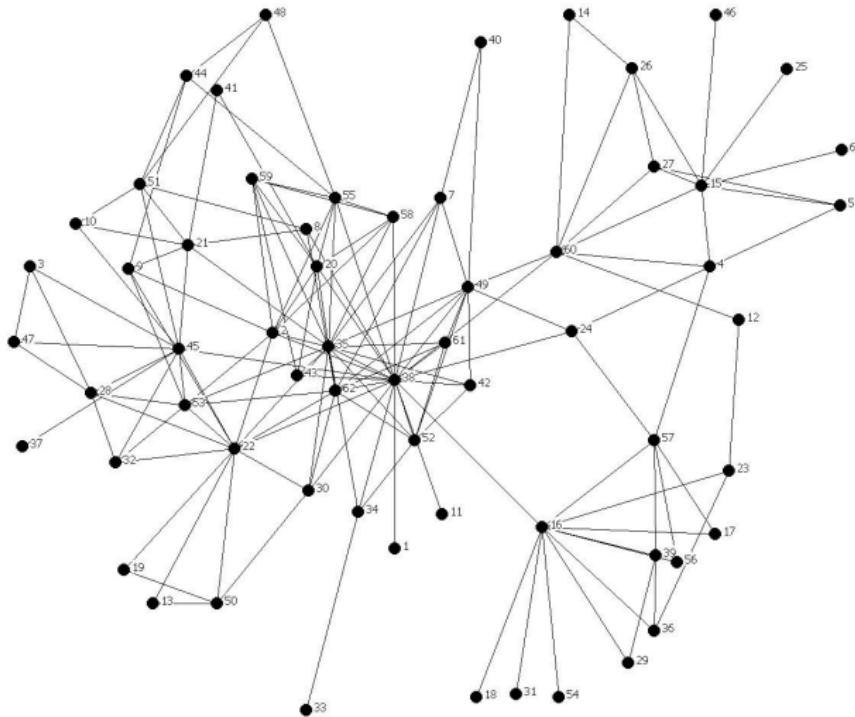


Results - Network Data from Krebs

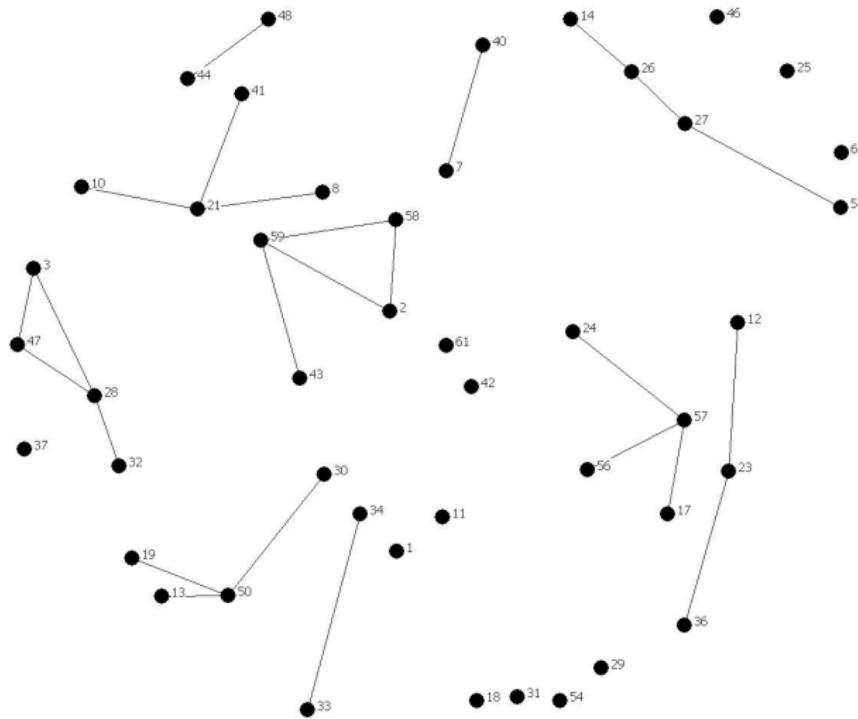


Heuristic found optimal solution in all cases in 0.02s on an average!

Krebs' terrorist network



Krebs' terrorist network



Results - Randomly Generated Sparse Graphs

Instance			IP Model		Heuristic		Heuristic + LS	
Nodes	Arcs	Deleted Nodes (k)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)
75	140	20	36	66.7	92	0.12	36	0.03
75	140	25	18	33.28	39	0.28	18	0.03
75	140	30	7	4.23	18	0.02	7	0.04
75	210	25	26	93.71	78	0.1	26	0.04
75	210	30	8	3.57	31	0.05	8	0.05
75	210	35	2	4.36	16	0.18	2	0.04
75	280	33	26	749.19	54	0.00	26	0.04
75	280	35	20	164.34	38	0.09	20	0.06
75	280	37	13	83.98	24	0.39	13	0.11
100	194	25	44	151.14	142	0.731	44	0.09
100	194	30	20	59.66	72	0.56	20	0.11
100	194	35	10	8.51	33	0.66	10	0.12
100	285	40	23	136.47	48	1.151	23	0.11
100	285	42	17	263.82	38	0.4	17	0.17
100	285	45	11	16.78	29	0.53	11	0.23

Results - Randomly Generated Sparse Graphs

Instance			IP Model		Heuristic		Heuristic + LS	
Nodes	Arcs	Deleted Nodes (k)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)	Obj Value	Comp Time (s)
100	380	45	22	128.13	58	0.58	22	0.15
100	380	47	16	243.07	42	1.191	16	0.16
100	380	50	10	228.72	23	0.31	10	0.11
125	240	33	62	5047.51	97	0.721	62	0.30
125	240	40	29	118.92	49	1.562	29	0.24
125	240	45	16	17.09	32	0.14	16	0.39
150	290	40	40	41.6	125	1.832	40	0.47
150	290	50	12	26.29	64	2.773	12	0.831
150	290	60	1	24.92	35	1.091	1	0.851
150	435	61	19	29.55	53	2.313	19	0.741
150	435	65	13	31.45	37	0.991	13	1.952
150	435	67	11	37.91	31	0.52	11	0.801

- ▶ Optimal solution found for each instance
- ▶ Average CPLEX time: 289.44 seconds
- ▶ Average Heuristic time: 0.33 seconds

What happened since 2009?

1. On new approaches of assessing Network vulnerability:
Approximation and hardness, Xuan, Thai, Pardalos, ACM, 2011
2. Robust optimization of graph partitioning and critical node detection, Fan, Pardalos, COA, 2010
3. Complexity of critical node detection problem in trees Di Summa et. al., C&OR, 2011
4. Multicriteria optimization model for humanitarian aid distribution, Vitoriano et. al, JOGO, 2011
5. Evaluation of strategies to mitigate contagion spread using social network characteristics, Ventresca, Aleman, SN, 2013
6. Polynomial algorithms for solving a class of critical node problems in trees and SP-graphs, Shen, Smith, Networks, 2012
7. Identifying critical nodes in PPI networks, Boginski, Commander, 2009
8. Economic analysis of N-k power grid contingency selection by interdiction methods, Fan, Xu, Pardalos, ES, 2011.
9. ...

Where are we headed?

CNP is as hopeless as ISP (from a worst case perspective)

Independent set problem

- ▶ ISP average case scenarios are tested
- ▶ A random graph $G(n, m)$, for $m = \frac{1}{2} \binom{n}{2}$,
 $\alpha(G(n, m)) \sim 2 \log_2(n)(1 + o(1))$ whp (tends to 1 as $n \rightarrow \infty$).
- ▶ A greedy inclusion maximal independent set is of size $\log_2 n$,
whp. [Grimmett and McDiarmid, 1975](#) (Still open: An $(1 + \epsilon) \log n$ -algorithm)
- ▶ For $d = \frac{2m}{n}$, $(2 - \epsilon_d)n \frac{\ln d}{d} \leq \alpha(G(n, m)) \leq (2 + \epsilon_d)n \frac{\ln d}{d}$ (with $\epsilon_d \rightarrow 0$ in the limit of large d)

Cardinality Critical Node Problem (CCNP)

- ▶ Alternate Formulation:
 - ▶ Suppose now, we want to limit the connectivity of the nodes.
 - ▶ 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
 - ▶ The property is hereditary and non-trivial (Yanakakis and Lund, 1999)
 - ▶ The problem is NP-hard and Max-SNP hard

CCNP - Heuristics

- ▶ We modified the MIS heuristic for this problem, but it is easy to create pathological instances.
- ▶ 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 - Network Data

Instance Max Conn. Index (L)	IP Model		Genetic Alg		ComAlg		ComAlg + LS	
	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

Optimal solution found for all Instances

Results - Computer Generated Instances

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

Results - Computer Generated Instances

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)
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

Solution Quality

- ▶ GA: optimal solutions found for 100% of cases
- ▶ MIS Heuristic: optimal solutions found for 87.5% of cases.

Results - Computer Generated Instances

Instance			Genetic Algorithm		ComAlg		ComAlg + LS	
Nodes	Arcs	Index (L)	Obj	Time (s)	Obj	Time (s)	Obj	Time (s)
75	140	5	18	1.622	21	0	18	1.502
	140	8	14	1.442	20	0.02	14	1.181
	140	10	12	1.231	20	0.12	12	3.364
75	210	5	23	1.532	29	0.01	23	18.476
	210	8	21	2.443	23	0.01	22	2.934
	210	10	20	2.794	24	0.09	20	21.17
75	280	5	31	3.464	35	0.101	31	3.144
	280	8	29	2.874	31	0.05	29	3.746
	280	10	28	3.775	30	0.13	28	4.787
100	194	5	22	5.317	33	0.02	22	2.774
	194	10	17	3.224	22	0.241	17	6.499
	194	15	15	2.954	22	0.021	15	0.44

Results - Computer Generated Instances

Instance			Genetic Algorithm		ComAlg		ComAlg + LS	
Nodes	Arcs	Index (L)	Obj	Time (s)	Obj	Time (s)	Obj	Time (s)
100	285	5	33	5.08	38	0.02	33	1.262
	285	10	28	4.376	31	0.05	28	11.076
	285	15	27	5.728	28	0.16	27	1.142
100	380	5	40	9.052	47	0.051	42	5.739
	380	10	36	11.506	41	0.02	37	3.866
	380	15	35	6.198	40	0.39	36	3.034
125	240	5	29	7.951	37	0.251	31	1.472
	240	10	24	9.984	29	0.07	24	1.993
	240	15	22	5.888	26	0.18	22	9.233
150	290	5	31	7.981	40	0.421	30	5.798
	290	10	26	4.967	32	0.2	25	5.107
	290	15	23	5.457	29	1.101	23	19.889
150	435	5	49	9.143	57	0.06	49	6.459
	435	10	40	19.407	50	0.44	41	5.518
	435	15	38	9.703	45	0.07	38	13.699

Formulation

min

$$\sum_{i \in V} v_i$$

s.t.

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

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

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

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

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

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

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

We have a better formulation!

$$\min \sum_{i \in V} x_i$$

$$\sum_{i \in S} x_i \geq 1, \quad \forall S \subset V, |S| = L + 1, G[S] \text{ is connected}$$

$$\mathbf{x} \in \{0, 1\}^{|V|}$$

Set cover formulation

We can get $\text{best}(O(L), O(\log n))$ -approximation using the LP relaxation

Hardness of approximation

- ▶ We can't do better than 2-approximation (from vertex cover)
- ▶ **Open problem:** Close this gap!

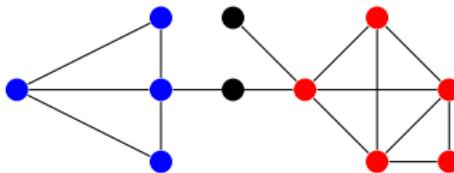
Also open: Fixed parameter tractability. Can we have an algorithm that runs in $f(k, L)O(n^c)$?

Densest k- subgraph

Problem definition

Input: Given a graph $G(V, E)$ and an integer k

Output: Find a set of k nodes, which induces the densest subgraph
(number of edges is maximised).



- ▶ The problem is NP-hard (even when the maximum degree is ≤ 3)
- ▶ It is APX-hard in general
- ▶ Best known approximation in the general case is $O(n)$

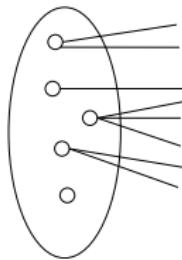
Bounded degree graphs (degree $\leq d$)

- ▶ APX-hard assuming the SSE conjecture is true
- ▶ $(1 - e^{\frac{1}{d}})$ -approximation is possible ($\approx \frac{1}{d}$)

Graph expansion

Given d -regular graph $G(V, E)$, expansion of a subset $S \subset V$ is

$$\Phi_G(S) = \frac{|E(S, V \setminus S)|}{d|S|},$$



Expansion of the graph with respect to $\delta > 0$:

$$\Phi_G(\delta) = \min_{|S|=\delta|V|} \Phi_G(S).$$

SSE Conjecture

Definition (Gap-SSE (η, δ))

Given a d -regular graph $G(V, E)$ and constants $\eta, \delta > 0$ distinguish whether:

Yes : $\Phi_G(\delta) \leq \eta$

No: $\Phi_G(\delta) \geq 1 - \eta$

The SSE conjecture is stated as follows:

Conjecture

For every $\eta > 0$, there exist a δ such that Gap-SSE (η, δ) problem is NP-hard.

[Raghavendra, Steurer, 2010]

APX-hard

- ▶ Intermediate problem: Find a node set S of size k that has the maximum value for $\frac{2|E(S)|}{dk}$
- ▶ “Same” as the DkS problem.
- ▶ Define the term $\bar{\Phi}_G(\delta) = 1 - \Phi_G(\delta)$
- ▶ This would distinguish instances
 - Yes : $\bar{\Phi}_G(\delta) \geq 1 - \eta$ and
 - No: $\bar{\Phi}_G(\delta) \leq \eta$.

Assuming that the SSE conjecture is true, this would yield a contradiction.

Greedy algorithm

Works with **weights** on nodes $w : V \rightarrow \mathbb{R}_+$ and **profits** on edges $p : E \rightarrow \mathbb{R}_+$. It also works on **hypergraphs** (with edges correspond to multiple nodes)

- ▶ Create new weights $w'_i = \frac{w_i}{d_i}$ for all nodes
- ▶ Arrange edges by profit to weight ratio $\frac{p_{i,j}}{(w'_i + w'_j)}$
- ▶ Enumerate all edge sets of size ≤ 2 ,
- ▶ Augment the set with edges in the above order (until budget B is not violated)

[Wolsey, 82], [Sviridenko, 03], [Khuller, Moss and Naor, 99]

A weak algorithm

$$\mathcal{P}_1 : \max \sum_{i=1}^m w_i x_i$$

$$\sum_{j=1}^n y_j \leq K$$

$$y_j \geq x_i, \forall (i,j) \in E$$

$$\mathbf{y} \in \{0,1\}^{|V|}$$

$$\mathbf{x} \in \{0,1\}^{|E|}$$

$$\mathcal{P}_2 : \max \sum_{i=1}^m w_i x_i$$

$$\sum_{j=1}^m w'_j y_j \leq \frac{K}{d}$$

$$\mathbf{y} \in \{0,1\}^{|V|}$$

- ▶ Every integral feasible solution to \mathcal{P}_2 is feasible to feasible to \mathcal{P}_1 .
- ▶ Every feasible solution to \mathcal{P}_1 scaled down by d is feasible to \mathcal{P}_1
- ▶ Solve \mathcal{P}_2 to get a $\frac{1}{d+\epsilon}$ -approximation for \mathcal{P}_1

Tighter Analysis of greedy

Let Y be the first two items we pick in our solution. Now consider the items in OPT and by the greedy algorithm $1, \dots, \ell, \ell + 1$. We define \widehat{W} as follows:

$$\widehat{W} = \sum_{e=1}^{\ell+1} W'_e = \sum_{e=1}^{\ell+1} \sum_{i \in S_e} \frac{w_i}{d_i}.$$

Multiplying by d throughout, we get

$$d\widehat{W} = \sum_{e=1}^{\ell+1} dW'_e = \sum_{e=1}^{\ell+1} \sum_{i \in S_e} d \frac{w_i}{d_i} \geq \sum_{i=1}^{\ell+1} \sum_{e=1}^{S_e} w_i \geq W(\bigcup_{e=1}^{\ell+1} S_e) \geq B.$$

Tighter Analysis

$$\begin{aligned}\sum_{j=1}^{\ell+1} p_i &\geq \left(1 - \prod_{j=1}^{\ell+1} \left(1 - \frac{W'_j}{B}\right)\right) P(OPT \setminus Y) \\ &\geq \left(1 - \prod_{j=1}^{\ell+1} \left(1 - \frac{W'_j}{d\widehat{W}}\right)\right) P(OPT \setminus Y) \\ &\geq \left(1 - \left(1 - \frac{1}{d(\ell+1)}\right)^{\ell+1}\right) P(OPT \setminus Y) \\ &\geq (1 - e^{-\frac{1}{d}}) P(OPT \setminus Y).\end{aligned}$$

Our solution = $P(Y) + \sum_{j=1}^{\ell} p_i$

Summary

- ▶ K-CNP
 - ▶ Integer linear programming
 - ▶ Complexity issues
 - ▶ Bicriteria results?
 - ▶ Algorithm for massive graphs?
- ▶ CC-CNP
 - ▶ Heuristics
 - ▶ Close the gap in approximation?
 - ▶ PTAS for planar graphs?
 - ▶ FPT algorithm?
- ▶ Greedy algorithm for DkS problem (bounded degree graphs)
 - ▶ Algorithms for massive graphs?

THANKS FOR YOUR ATTENTION!