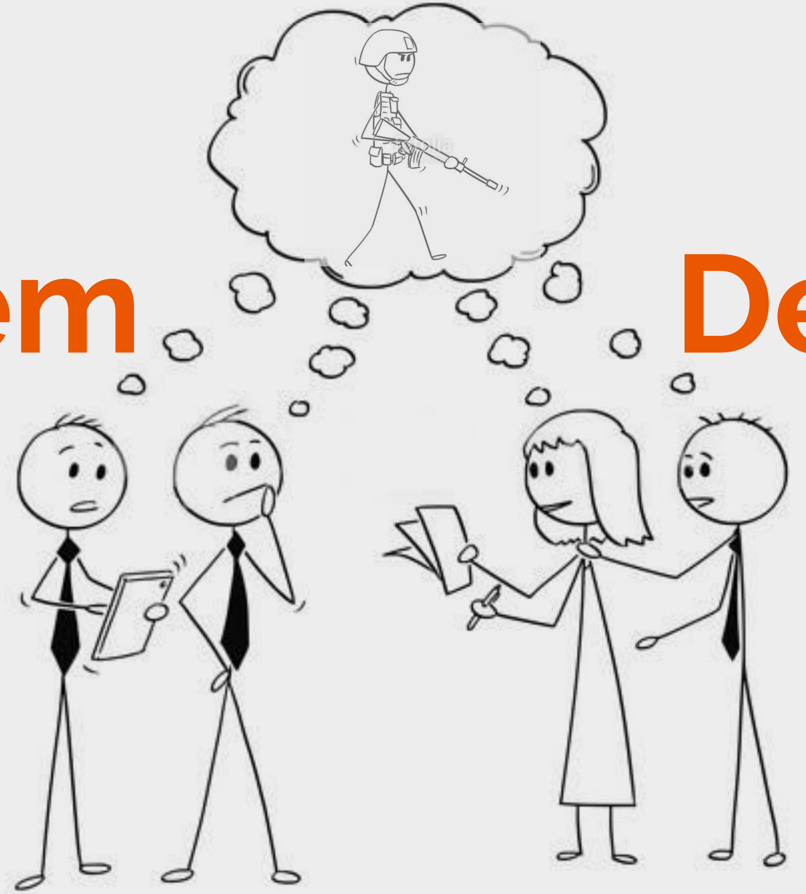
On Parameterized Complexity of **Network Modifications** for **Binary Networked Public Goods Games**

B.Tech Project
Amatya Sharma

Mentor : Dr. Palash Dey
Indian Institute of Technology Kharagpur

NOT continued from the work of BTP-1 . . .

Problem Definition



BNPG (Binary Networked Public Goods) Games

Given:

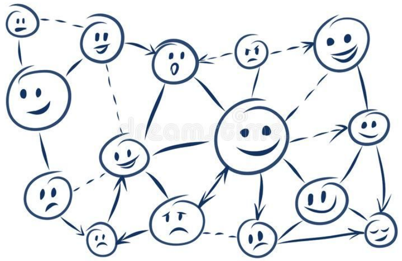
- Network as Undirected graph with players as vertices
- Each player i can either invest ($x_i = 1$) or not ($x_i = 0$)
- Utility of i^{th} player :

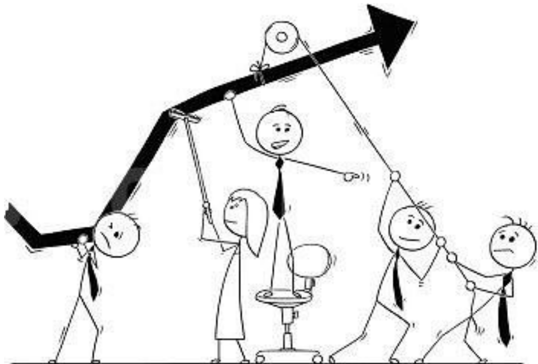
$$\boxed{U_i(x) = U_i(x_i, n_i^x) = g_i(x_i + n_i^x) - c_i x_i}$$

where :

- n_i^x := #neighbors investing
- $g_i(.)$:= non − negative non − decreasing

x_i := Strategy played by i^{th} player
x = (x1, ..., xn) := Joint pure strategy profile of all players

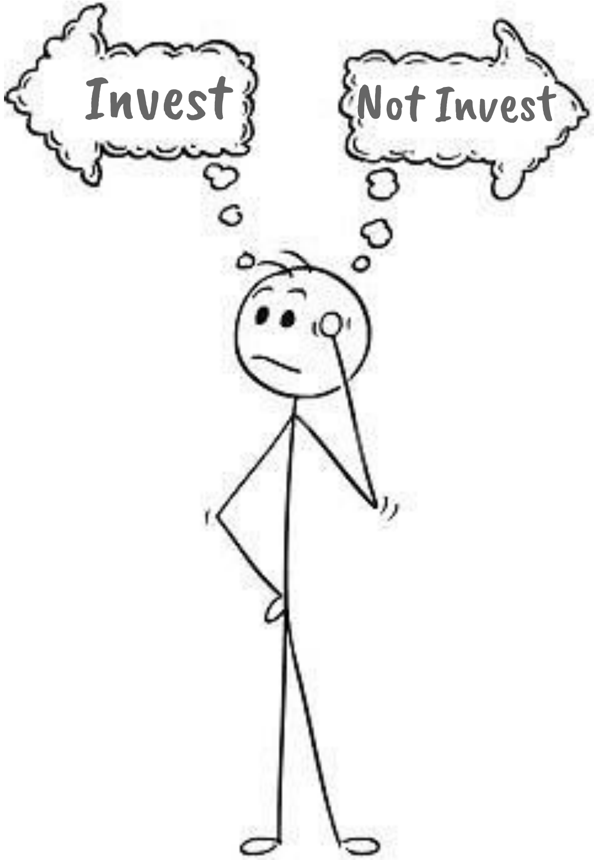




PSNE (Pure Strategy Nash Equilibria) of BNPG Games

A Joint Pure Strategy Profile $x \in \{0, 1\}^n$ such that:

- $U_i(x_i, n_i^x) > U_i(1 - x_i, n_i^x)$, or
- $U_i(x_i, n_i^x) = U_i(1 - x_i, n_i^x)$ and xi = 1



Who Invests?? PSNE Classes

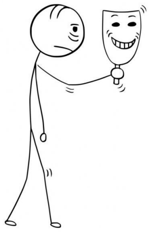
all: every player invests i.e. $x = (1, 1, \ldots, 1)$

$= S$: only set S invests

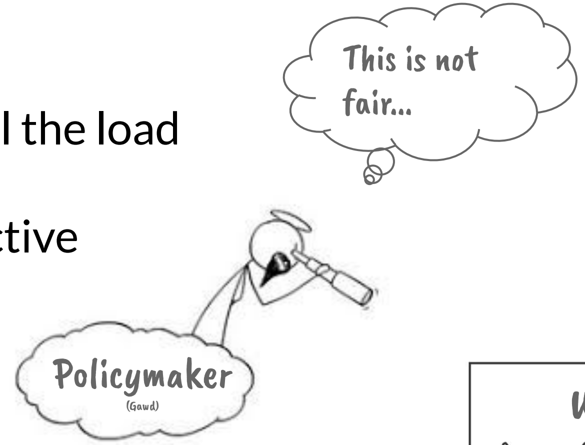
$\supseteq S$: superset of set S invests

$\geq r$: at least r players invest

What's the Problem then???



- A few "diligent" workers may bear all the load
- Detrimental for a long-term perspective
- Turns out to be unfair



Not ENOUGH to find PSNE of BNPG

Network Modifications : Tackling Unfairness

A central mechanism (algorithm) ensuring:

- A specified set of players invest
- Break existing connections (delete edges)
- Make new connections (add edges)
- Bribe them!!!

Edge Edition!!
Addition
+
Deletion

$g_i(\cdot)$: what forms it can take?

- Captures how a player behaves w.r.t increasing investment of its neighbors
- Non − negative, Non − decreasing

Can be :

- general
- convex (increasing returns)
- concave (diminishing returns)
- sigmoid (first increasing then diminishing returns)

Investment Degree Set (D_i)

A unique set $D_i \subseteq \{0, 1, ..., n-1\}$ such that:

- $x_i = 1$ is a best response $\Leftrightarrow$ $n_i^x \in Di$

Interesting property:

- g_i is concave $\Leftrightarrow$ D_i is downward-closed interval
- g_i is convex $\Leftrightarrow$ D_i is upward-closed interval
- g_i is sigmoid $\Leftrightarrow$ D_i is an interval

NDDS(P,X) (Network Design for Degree Sets)

Given :

- BNPG instance := (Graph & utilities $U_{i \in [n]}$)
- D_i := investment degree sets for all players $i \in [n]$
- $\gamma_{e \in nC2}$:= Edge costs
- X := desired PSNE class (all, $= S$, $\supseteq S$, $\geq r$)
- P := Property of $g_i(\cdot)$ (convex, concave, sigmoid, or general)
- k := budget k

Goal :

Decide whether there exists an edge set S with:

- $\sum_{e \in E \ominus S} \gamma_e \leq k$
- $\exists$ I $\in$ X of investing players such that in the modified graph G' (V, E' = E $\ominus$ S)

$$|N_i^{G'} \cap I| \in D_i \quad \forall i \in I$$
$$|N_i^{G} i \cap I| \notin D_i \quad \forall i \notin I.$$

No Budget !! (k=0)

$\gamma_{e \in nC2} > 0$

NDDS reduces to :

- Finding PSNE for BNPG
- Without any modifications allowed

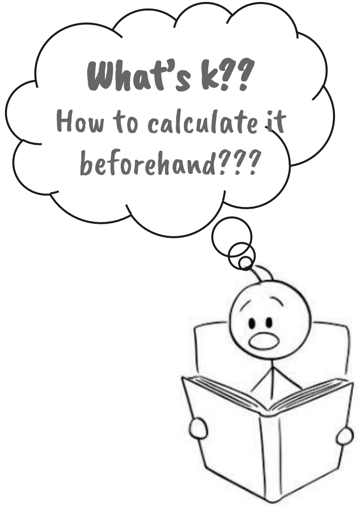
Preliminaries

Parameterized Algorithms

Parameterized problem : Language L ⊆ $\Sigma^* \times N$, where Σ is a fixed, finite alphabet. For an instance $(x, k) \in \Sigma^* \times N$, k is called the parameter.

Parameterized Algorithms

- **Fixed Parameter Tractable [FPT] :** Running time $f(k)n^c$
- **Slice-wise Polynomial [XP] :** Running time $f(k)n^{g(k)}$



Para-NP-Hardness

Para-NP [Flum and Grohe]:

Class of parameterized problems solvable in time $\mathbf{f(k).|x|^{O(1)}}$ by a nondeterministic TM [f is computable]

Para-NP-Hard :
NP-Hard for a constant value of parameter

NP-Hard for a "Slice" of the parameter

W[t]-Hardness

W[t]

If there is a parameterized reduction from problem P to WCS[$C_{t,d}$] for some d > 1

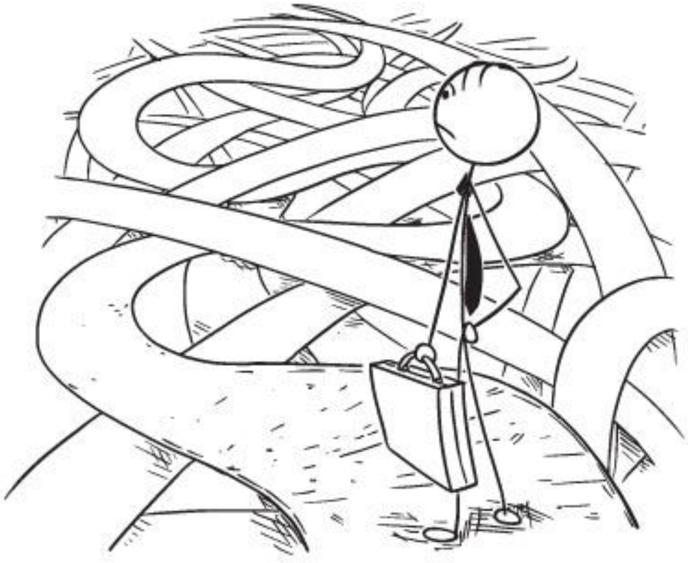
W[t]-Hard :
If every problem in W[t] can be reduced to P

WCS??
Weighted Circuit Satisfiability

FPT

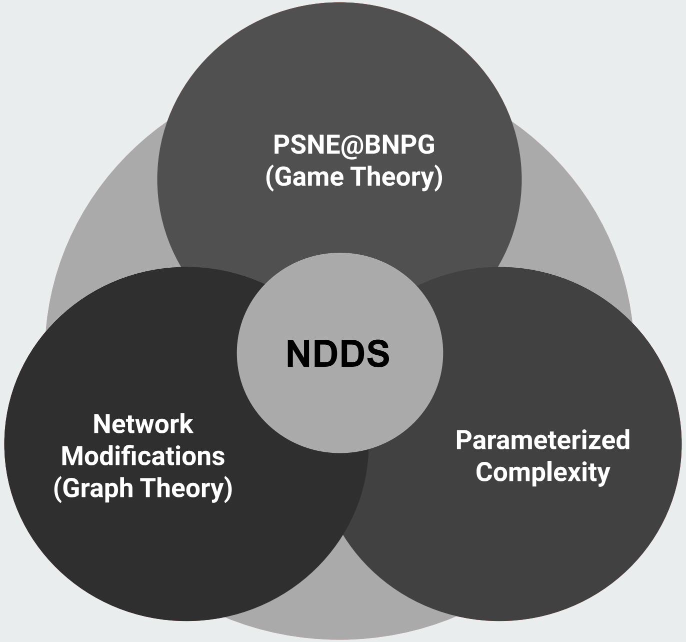
Parameters Under Consideration

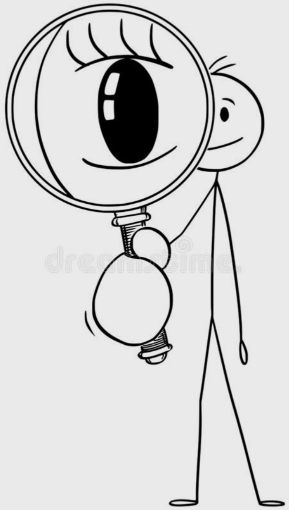
- k := input budget
- r := NDDS (P, **r**)
- α := $\min_{v \in V[G]}$ lower bound(D_v)
- δ := diameter of input graph
- n_U := number of distinct utility functions
- tw := treewidth of graph*
- D := $\max_{v \in V[G]} |D_v|$
- Δ := max degree of input graph



Skipping over the Prior Results . . .

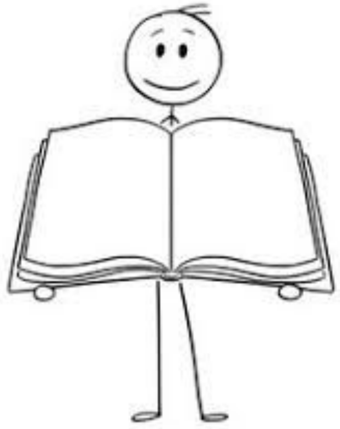
Our Results

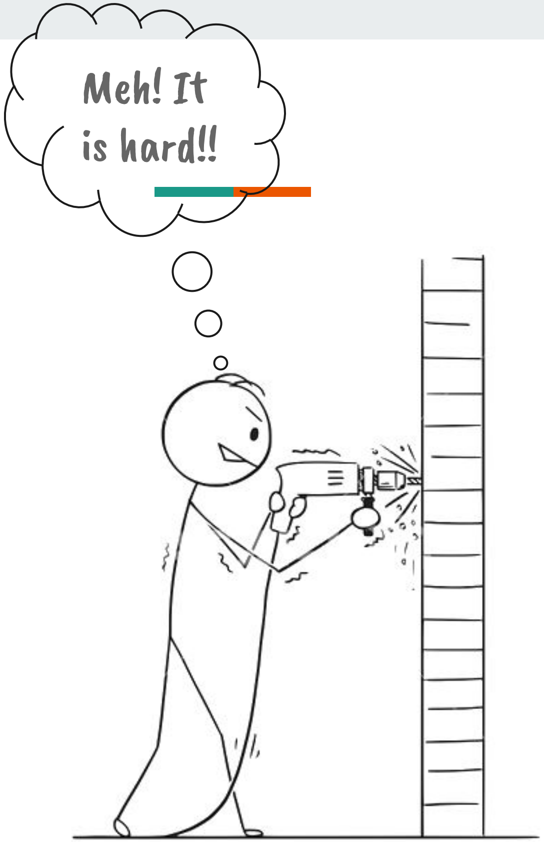




Summary of Our Results

Problem Variant	Parameter	Result
all, general	k (budget)	W[1]-Complete Theorem 15
$\{= S,\ \supseteq S,\ \geqslant r\}$, general	k	W[1]-Complete Theorem 16
$\{\supseteq S,\ \geqslant r\}$, concave	k	W[1]-Complete Theorem 17
$\{\supseteq S,\ \geqslant r\}$, sigmoid	k	W[1]-Complete Theorem 18
$\geqslant r$, { concave, convex, sigmoid}	$r + k$	W[1]-Complete Theorem 19
$\geqslant r$, convex	$k + r + \alpha$	W[1]-Hard Theorem 23
$\geqslant r$, sigmoid	r	para-NP-hard Theorem 25
$\{\geqslant r,\ \supseteq S\}$, general	$\lvert I \rvert$	W[2]-Hard Observation 2
$\{\geqslant r,\ \supseteq S\}$, general	$n - \lvert I \rvert$	W[2]-Hard Observation 2
$\{\geqslant r,\ \supseteq S\}$, general	treewidth	W[1]-Hard Observation 3
$\{\geqslant r,\ \supseteq S\}$, general	Δ	para-NP-hard Observation 4
$\{\geqslant r,\ \supseteq S\}$, general	(δ, n_u)	para-NP-hard Observation 6, Observation 5
{−any−, −any−}, -any-	k	$n^{O(k)}$ XP algorithm Observation 7
$\{\geqslant r\}$, convex	r	$n^{O(r^2)}$ XP algorithm Theorem 26





Hardness Results

Result1 : NDDS (general, all) - W[1]-C w.r.t k

Thm. The problem of NDDS (general, all) is W[1]−Complete w.r.t the parameter k (budget).

Even when the input graph is unweighted

Black Box

r-regular Clique
W[1]-Complete

+

Reduction

r-regular Clique

↓

NDDS
(general, all)

=

NDDS
(general, all)

W[1]-C

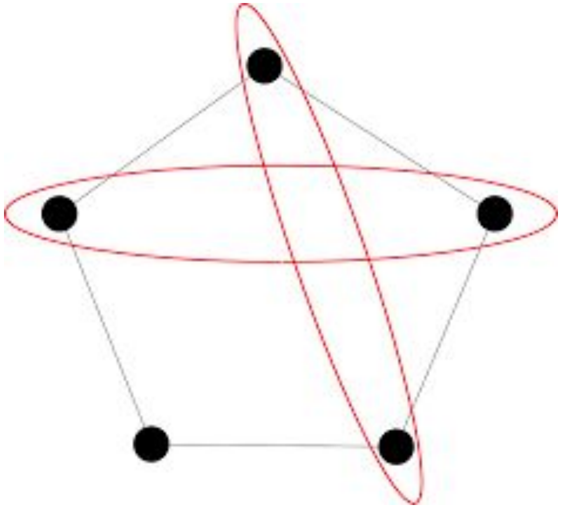
w.r.t the
parameter k

r-regular Clique

Input : (G(V, E), k)

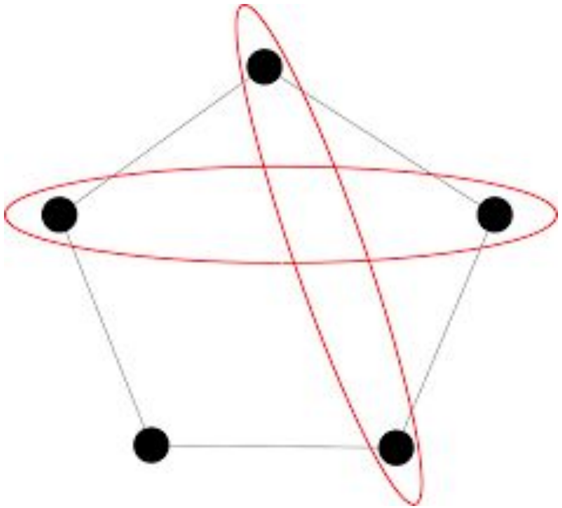
➢ G is r-regular undirected graph

Goal : Decide whether there exists a k-clique as a subgraph of G



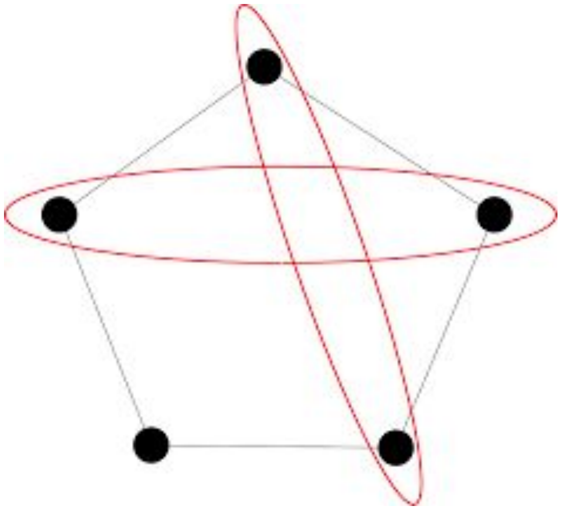
Main Reduction

- $\triangleright$ $V'[G'] = V[G] \cup Z$, where $Z = \{z_1, ..., z_k\}$;
- $\triangleright$ $E'[G'] = E[G] \cup \{(v_i, z_j) \mid \forall v_i \in V[G], \ , j \in [k]\}$;
- $\triangleright$ $\gamma_e = 1, \quad \forall e \in E'[G']$;
- $\triangleright$ $D_{v_i} = \{r - k - 1, \ r + k\}, \quad \forall v_i \in V[G]$;
- $\triangleright$ $D_{z_j} = \{n - k\}, \ \forall j \in [k]$;
- $\triangleright$ $k' = k^2 + \binom{k}{2}$.



A Corollary...

- Set $S = V[G]$ and $r = n$ accordingly
- W[1] – Completeness of
  - NDDS (general, =S)
  - NDDS (general, $\supseteq$ S)
  - NDDS (general, $\geq$ r)
    - with respect to the parameter **k.**



Result2 : NDDS (convex, ≥ r) - W[1]–C w.r.t (k + r + α)

Thm. NDDS (convex, ≥ r) is W[1]-hard with respect to the parameter k + r + α.

W[1]-hard w.r.t parameter k+r even when α = 3 even when the graph is unweighted.

Black Box

Edge-k-Core
W[1]-Hard
w.r.t k+r (α = 3)

+

Reduction

Edge-k-Core

$\Downarrow$

NDDS
(convex, ≥ r)

=

NDDS

(convex, ≥ r)

W[1]-H

w.r.t

k + r + α

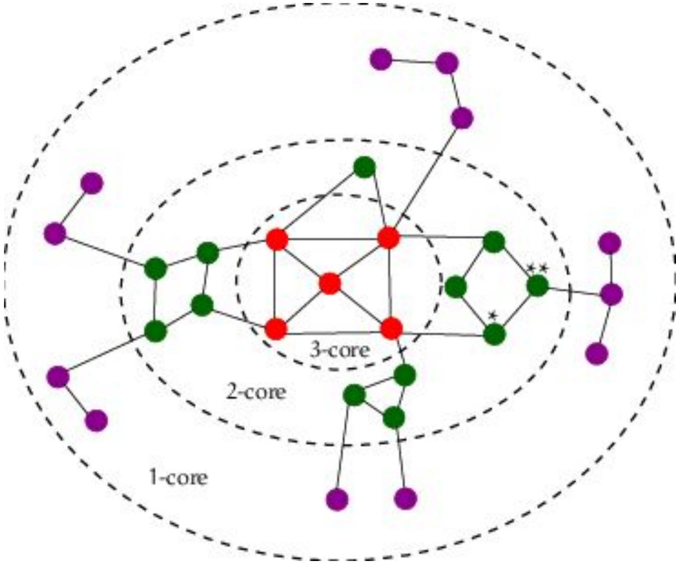
Edge-k-Core

Input : (G(V, E), k)

- Simple, undirected graph G = (V, E)
- Integers k, α, and r

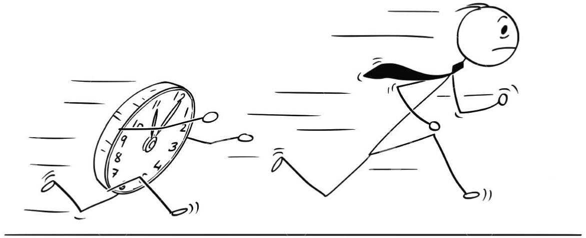
Goal : Decide if there exists $H \subseteq V[G]$ such that:

- Adding at most k edges to G
- In modified graph G', every $v \in H$ has $\deg_{G'[H]}[v] \geq \alpha$



Main Reduction

1. $G^* = G$ i.e. $V^* = V$ and $E^* = E$;
2. $D_v = \{\alpha, \ \ldots, \ n-1\} \ \forall v \in V^*$;
3. $r^* = r$
4. $k^* = k$



Result3 : NDDS (sigmoid, ≥ r) - para-NP-hard w.r.t r

Thm. NDDS (sigmoid, ≥ r) is para-NP-hard w.r.t parameter r

even when $\max(|D_v|) = 1$, $k=0$, and the graph is unweighted

Black Box

r-regular Subgraph
para-NP-hard
w.r.t r

+

Reduction

r-regular Subgraph

↓

NDDS
(sigmoid, ≥ r)

=

NDDS

(sigmoid, ≥ r)

para-NP-hard

w.r.t

r

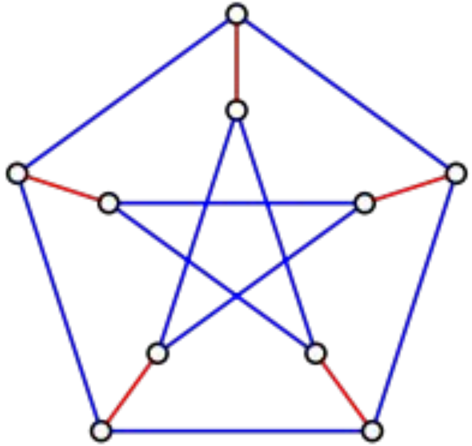
r-regular Subgraph

Input : (G(V, E), r)

- Simple, undirected graph G = (V, E)
- Positive Integer r

Goal : Decide whether there exists a H ⊆ V[G], such that-

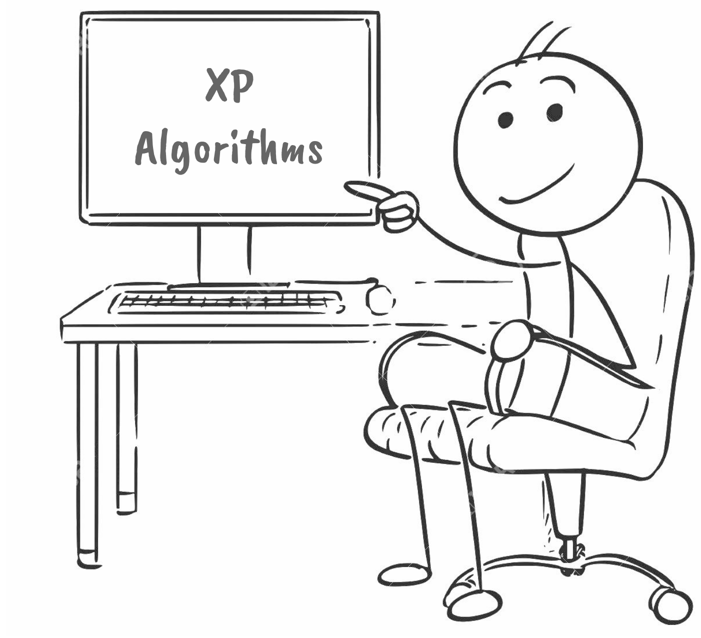
- Subgraph G[H] is r-regular



Idea of Reduction

1. $G^* = G$ i.e. $V^* = V$ and $E^* = E$;
2. $D_v = \{r\}\ \forall v \in V^*$;
3. $r^* = r$
4. $k^* = 0$
5. weight of each edge $= 1$.

Algorithmic Results



Result4 : XP w.r.t k

Thm.

All versions of NDDS can be solved in XP time $n^{O(k)}$

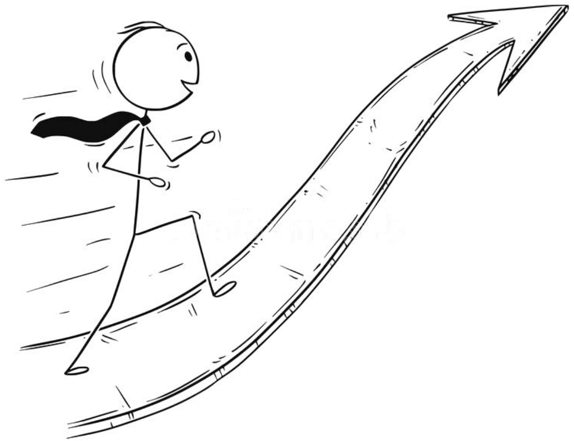
We already:

- Established W[1]-Completeness results w.r.t k
- Ruling out any FPT-Algorithm
- Designed the next best : XP

Result6 : NDDS (concave, ≥ r) - XP w.r.t r

Thm.

NDDS (convex, ≥ r) can be solved in XP time $n^{O(r.r)}$.



We already:

- Established W[1]-Completeness results w.r.t r
- Ruling out any FPT-Algorithm
- Designed the next best : XP

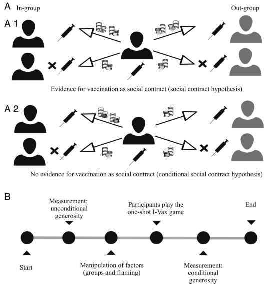
Conclusions & Significance of Our Work

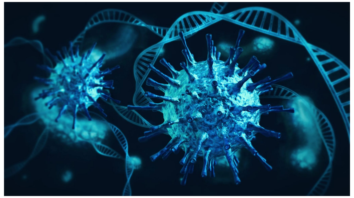
- Notched up the results taking into account the parameterized complexity w.r.t key natural as well as structural parameters
- Crucial role in computer science, economics, game theory and network design
- Lower Bound by W[1]-hardness
- Ruling out FPT
- Upper bound by XP-algorithms, making the analysis complete

Future Directions

- ➢ Approximate, i.e., ε-PSNE for the problem...
- ➢ More structural parameters like VC, FVS, FAS...
- ➢ Problem formulation on line-graph of the input graph...
- ➢ XP algorithms w.r.t treewidth or maximum degree...
- ➢ Trivial graph classes like trees/forests, cycles, paths, caterpillars...
- ➢ Color coding
- ➢ Parameterization by distance to trees, paths or cluster graphs...

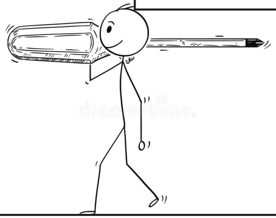
Practical Implications



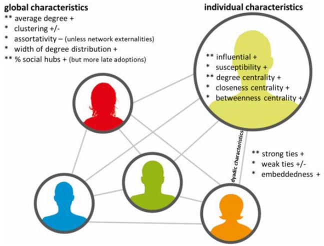


- Modeling **Behavioral Response to Vaccination** Using Public Goods Game *by Ben-Arieh et al.*
- Vaccination as a **Social Contract** *by Korn et al.*

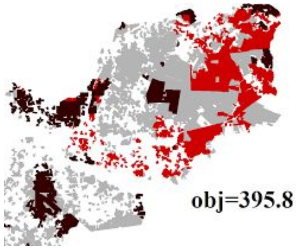
Game Theory of **Social Distancing** in **Response to an Epidemic** *by Rulega*



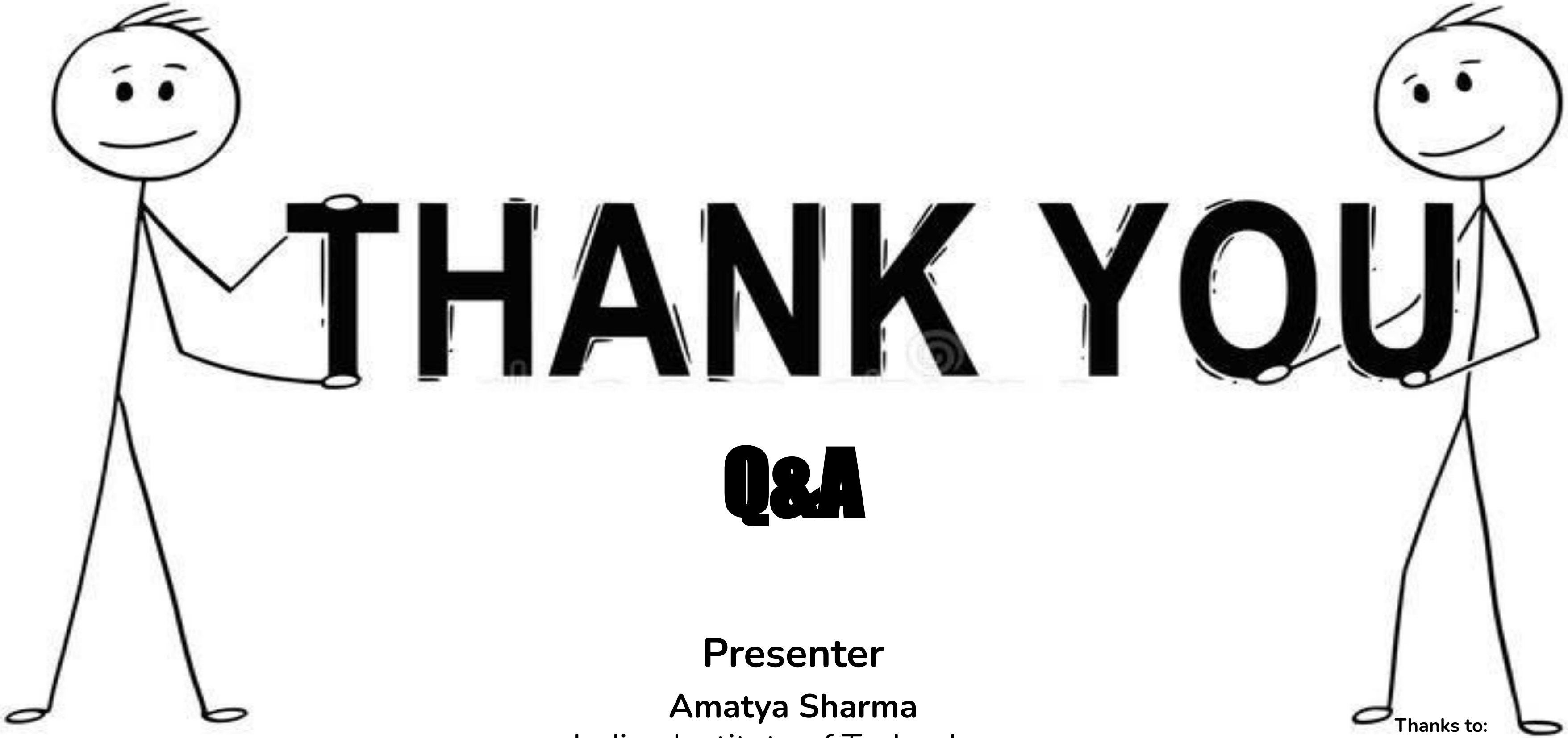
Manipulating **opinion diffusion** in **social networks** *by Bredereck et al.*



Election Control in Social Networks using **Edge edition** by *Castiglioni et al.*



Maximizing **spread of cascades** using **Network Design** by *Sheldon et al.*



Q&A

Presenter

Amatya Sharma
Indian Institute of Technology
Kharagpur

Thanks to:
Aay
StickMIT, Stickassachusetts

Result4 : NDDS (concave, {⊇ S, ≥ r}) - W[1]−C w.r.t k

Thm. NDDS (concave, X) for X ∈ {⊇ S, ≥ r} is W[1] – Complete w.r.t the parameter k.

Black Box

Independent Set
W[1]-Complete

+

Reduction

Independent Set

↓

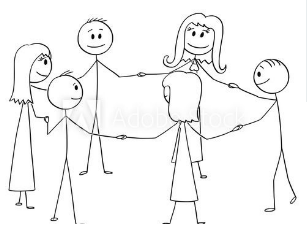
NDDS (concave, {⊇ S, ≥ r})

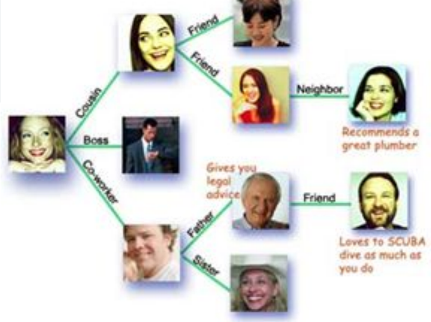
=

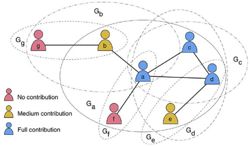
NDDS (concave, {⊇ S, ≥ r})

W[1]-C

w.r.t the parameter k




Friend
Friend
Cousin
Neighbor
Boss
Recommends a great plumber
Co-worker
Gives you legal advice
Friend
Father
Sister
Loves to SCUBA dive as much as you do


G_b
G_g
G_c
G_a
G_f
G_e
G_d
No contribution
Medium contribution
Full contribution