Super-polynomial time approximability of inapproximable problems

Joint work with Michael Lampis and Vangelis Paschos

FPT weekly seminar

July 9, 2015

Introduction

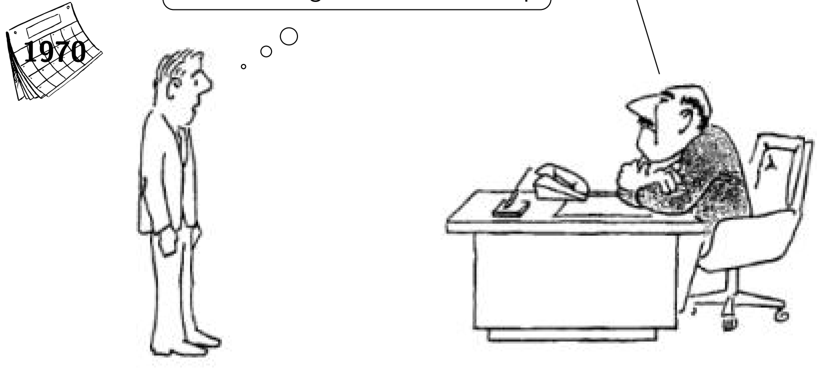
Min Independent Dominating Set

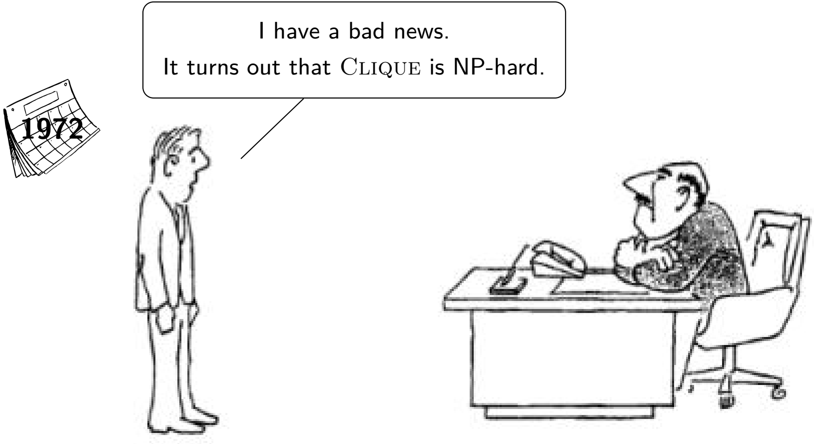
Max Minimal Vertex Cover

Max Induced Path/Forest/Tree

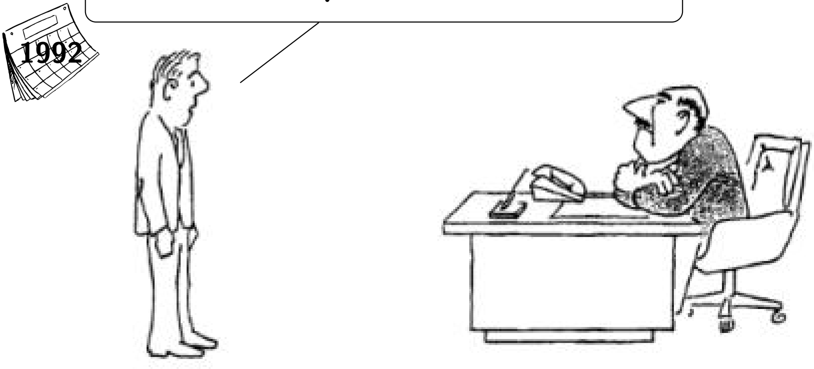
Can you find an efficient algorithm for CLIQUE?

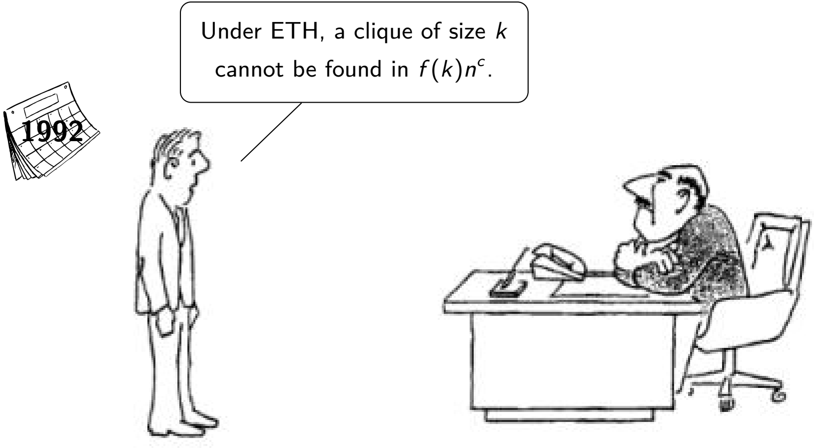
Phew! For a moment, I thought he would bring BANDERSNATCH up.



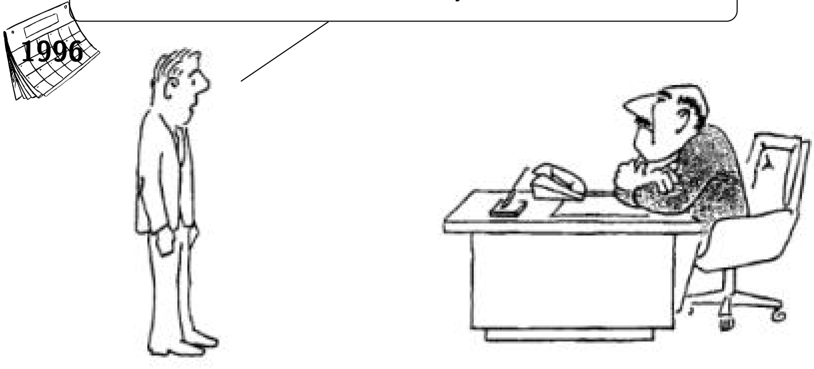

1972
I have a bad news.
It turns out that CLIQUE is NP-hard.

Under P $\neq$ NP, there is no polytime n^{ε}-approximation for CLIQUE for some $\varepsilon > 0$.

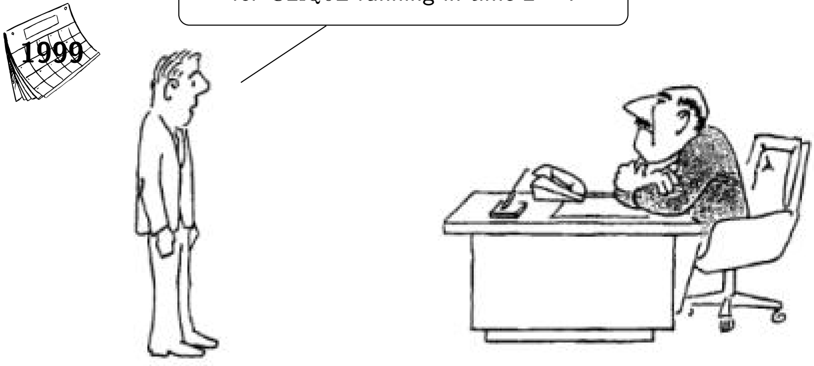



Under ETH, a clique of size k cannot be found in $f(k)n^c$.
1992

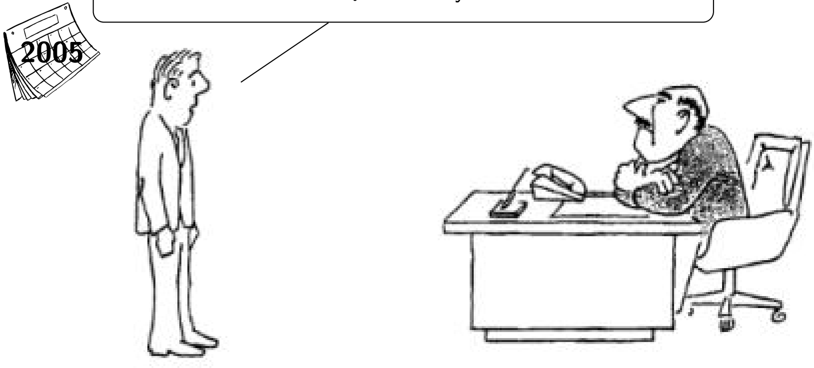
Under NP $\neq$ co-RP, there is no polytime $n^{1-\varepsilon}$-approximation for CLIQUE for any $\varepsilon > 0$.

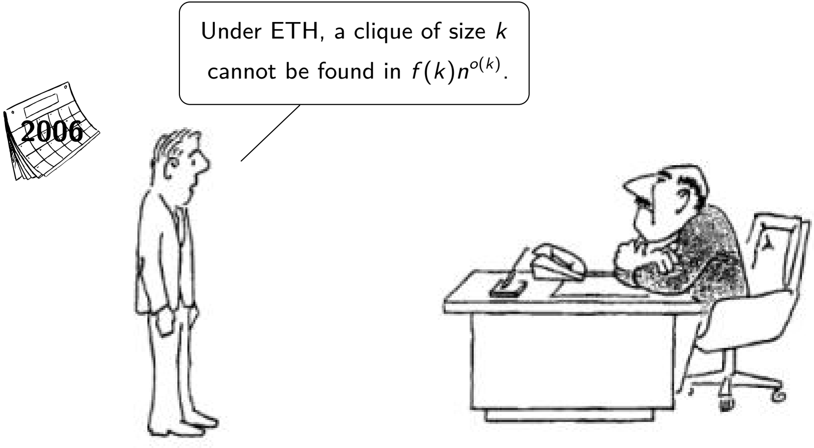


Under ETH, there is no exact algorithm for CLIQUE running in time $2^{o(n)}$.



Under P $\neq$ NP, there is no polytime $n^{1-\varepsilon}$-approximation for CLIQUE for any $\varepsilon > 0$.




2006
Under ETH, a clique of size k cannot be found in $f(k)n^{o(k)}$.

What can we do then?

Can we $n^{\frac{1}{100}}$-approximate $\textsc{Clique}$ in time $n^{\log n}$? 

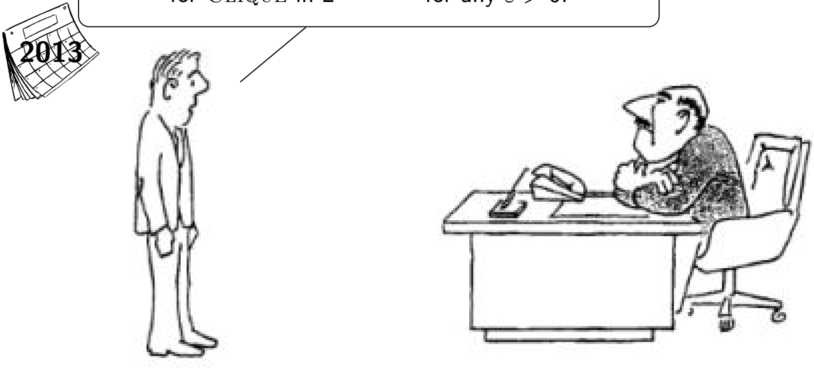
What can we do then?

Can we $\log^* n$-approximate $\textsc{Clique}$ in time $n^{\alpha(n)}$?

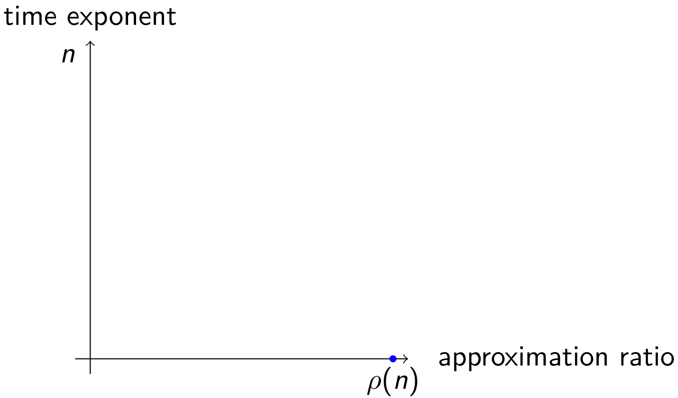
What can we do then?

Can we $\sqrt{n}$-approximate CLIQUE in time $2^{\sqrt{n}}$?

Under randomized ETH, there is no r-approximation for CLIQUE in $2^{n^{1-\varepsilon}/r^{1+\varepsilon}}$ for any $\varepsilon > 0$.

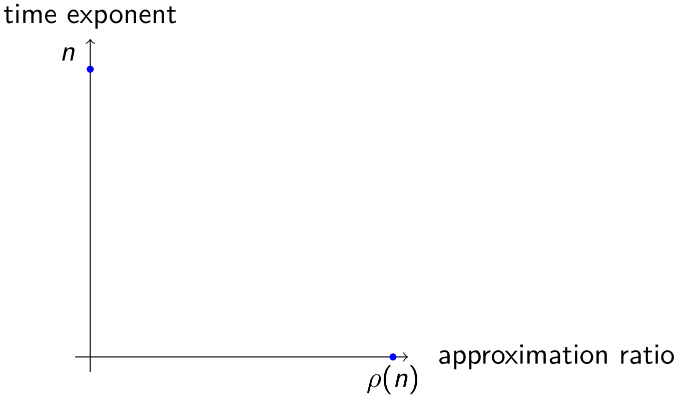


What we aim



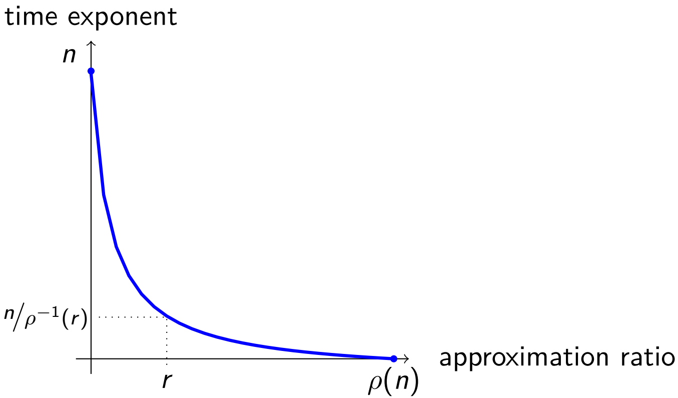
Π is polytime $\rho(n)$-approximable.

What we aim



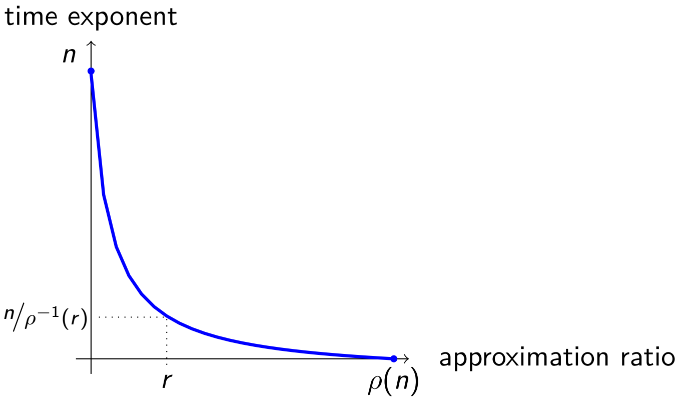
Π is exactly solvable in time λ^n.

What we aim



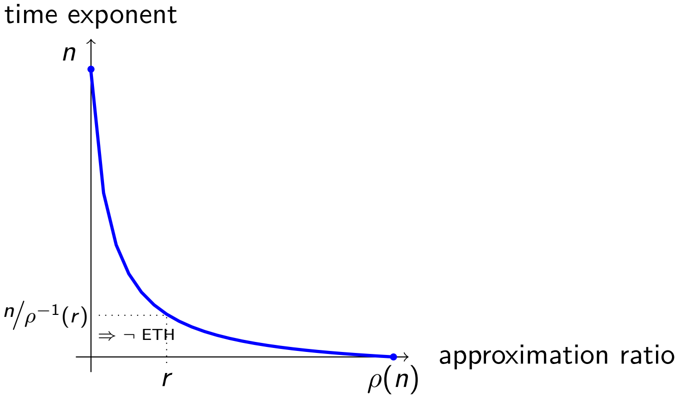
Can we show that Π is $\rho(r)$-approximable in time $\lambda^{n/r}$?

What we aim



Or, at least, $\rho(r)$-approximable in time $2^{n \log r/r}$?

What we aim



And then show almost matching lower bounds under ETH?

About neglecting polylog factors

For a graph with 10000 vertices and a clique of size 5000:

algorithm	1.1996^n (XN '13)	ratio $\sqrt{n}$ in $1.1996^{\sqrt{n}}$	ratio $\frac{n(\log\log n)^2}{(\log n)^3}$ (F '05)
limit	100 vertices	10000 vertices	1000000 vertices
output	don't ask	a clique of size > 50	

About neglecting polylog factors

For a graph with 10000 vertices and a clique of size 5000:

algorithm	1.1996^n (XN '13)	ratio $\sqrt{n}$ in $1.1996^{\sqrt{n}}$	ratio $\frac{n(\log\log n)^2}{(\log n)^3}$ (F '05)
limit	100 vertices	10000 vertices	1000000 vertices
output	don't ask	a clique of size > 50	a clique of size > 50

Subset problems

- Solutions are subsets of vertices (edges, elements, sets).

Subset problems

- Solutions are subsets of vertices (edges, elements, sets).
- Monotonicity:
  - Minimization: S feasible $\Rightarrow \forall T \supseteq S$, T feasible.
  - Maximization: S feasible $\Rightarrow \forall T \subseteq S$, T feasible.

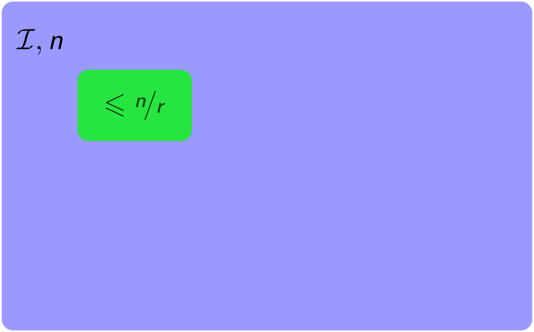
Subset problems

- Solutions are subsets of vertices (edges, elements, sets).
- Monotonicity:
  - Minimization: S feasible $\Rightarrow \forall T \supseteq S$, T feasible.
  - Maximization: S feasible $\Rightarrow \forall T \subseteq S$, T feasible.
- Weak monotonicity:
  - Minimization: S feasible $\Rightarrow \exists v \notin S$, $S \cup \{v\}$ feasible.
  - Maximization: S feasible $\Rightarrow \exists v \in S$, $S \setminus \{v\}$ feasible.

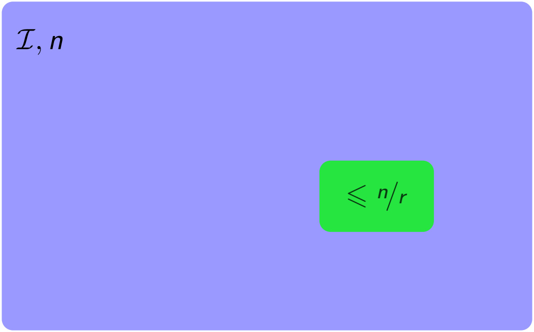
Minimization subset problems

$\mathcal{I}, n$

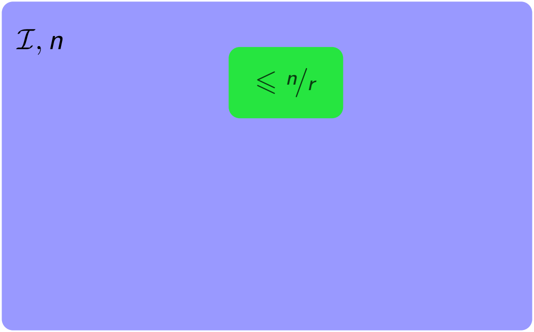
Minimization subset problems



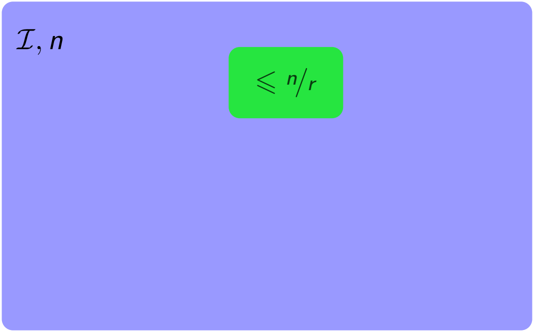
Minimization subset problems



Minimization subset problems

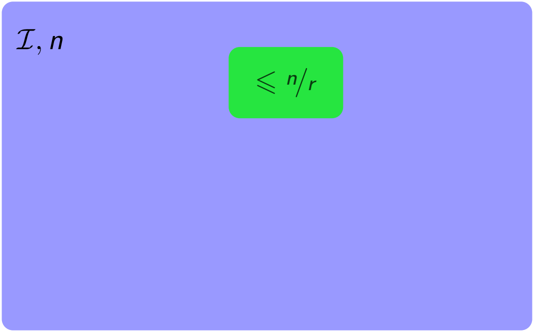


Minimization subset problems



- If a solution is found, it is an optimal solution.

Minimization subset problems



- If a solution is found, it is an optimal solution.
- If not, any feasible solution is an r-approximation.

Weakly monotone maximization subset problems

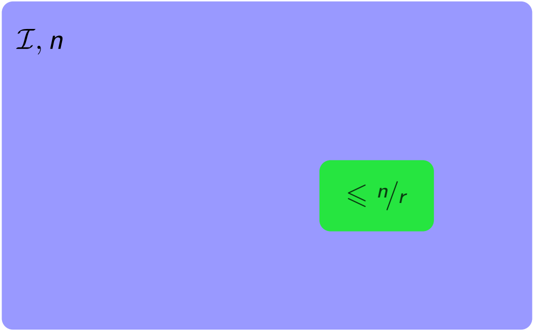
$\mathcal{I}, n$

Weakly monotone maximization subset problems

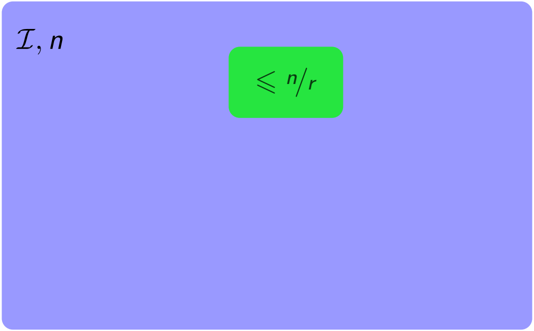
$\mathcal{I}, n$

$\leqslant n/r$

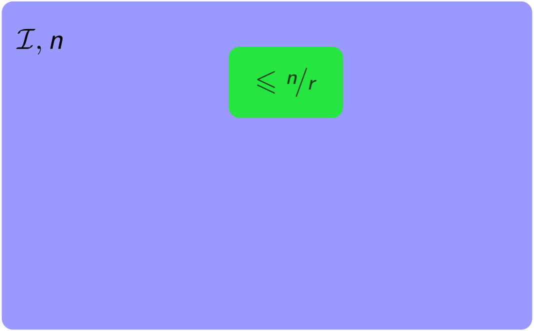
Weakly monotone maximization subset problems



Weakly monotone maximization subset problems

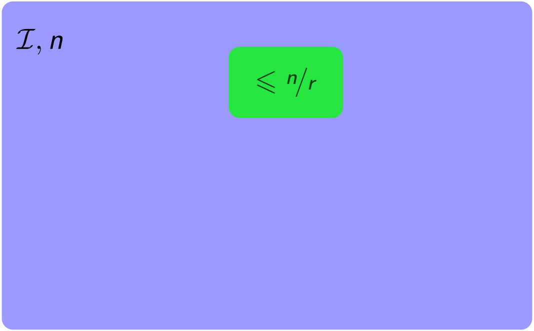


Weakly monotone maximization subset problems



- If a solution is found, it is an r-approximation.

Weakly monotone maximization subset problems



- If a solution is found, it is an r-approximation.
- If not, there is no feasible solution.

The r-approximation takes time
$O^*(\binom{n}{n/r}) = O^*((\frac{en}{n/r})^{n/r}) = O^*((er)^{n/r}) = O^*(2^{n\log(er)/r})$.

The r-approximation takes time
$O^*(\binom{n}{n/r}) = O^*((\frac{en}{n/r})^{n/r}) = O^*((er)^{n/r}) = O^*(2^{n\log(er)/r})$.

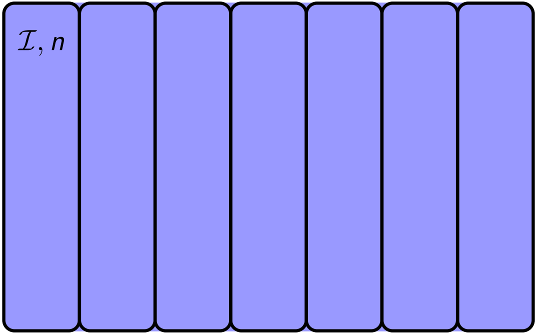
Can we improve this time to $O^*(2^{n/r})$?

The r-approximation takes time
$O^*(\binom{n}{n/r}) = O^*((\frac{en}{n/r})^{n/r}) = O^*((er)^{n/r}) = O^*(2^{n\log(er)/r})$.

Can we improve this time to $O^*(2^{n/r})$?

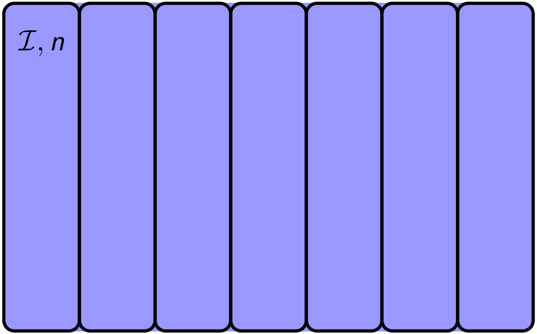
- Splitting
- Merging

Splitting: monotone maximization subset problems



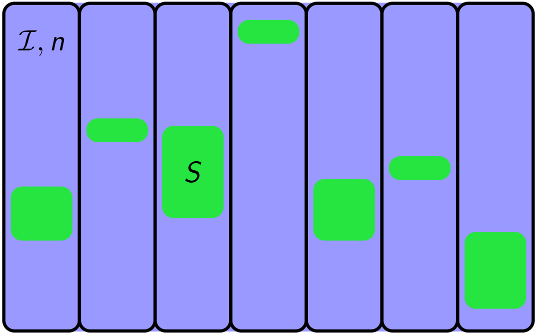
Split the instance into r parts of size n/r.

Splitting: monotone maximization subset problems



For each of the $r2^{n/r}$ subsets of each part, check the feasibility.

Splitting: monotone maximization subset problems



Fix an optimal solution. The output is at least as good as S.

Merging: Set Cover

- $\mathcal{I} = \{S_1, S_2, \ldots, S_m\} \rightsquigarrow \mathcal{I}' = \{S'_1 = S_1 \cup \ldots \cup S_r,$ $S'_2 = S_{r+1} \cup \ldots \cup S_{2r}, \ldots, S'_{m/r} = S_{m-r+1} \cup \ldots \cup S_m\}$.

Merging: Set Cover

- $\mathcal{I} = \{S_1, S_2, \ldots, S_m\} \rightsquigarrow \mathcal{I}' = \{S'_1 = S_1 \cup \ldots \cup S_r, S'_2 = S_{r+1} \cup \ldots \cup S_{2r}, \ldots, S'_{m/r} = S_{m-r+1} \cup \ldots \cup S_m\}$.
- Solve optimally $\mathcal{I}'$ in time $2^{m/r}$ $\rightarrow$ solution $S'_{a_1}, \ldots, S'_{a_k}$.

Merging: Set Cover

- $\mathcal{I} = \{S_1, S_2, \ldots, S_m\} \rightsquigarrow \mathcal{I}' = \{S'_1 = S_1 \cup \ldots \cup S_r,$
  $S'_2 = S_{r+1} \cup \ldots \cup S_{2r}, \ldots, S'_{m/r} = S_{m-r+1} \cup \ldots \cup S_m\}$.
- Solve optimally $\mathcal{I}'$ in time $2^{m/r}$ $\rightarrow$ solution $S'_{a_1}, \ldots, S'_{a_k}$.
- Take all the sets of $\mathcal{I}$ composing the S'_{a_i}.

Min Asymmetric Traveling Salesman Problem

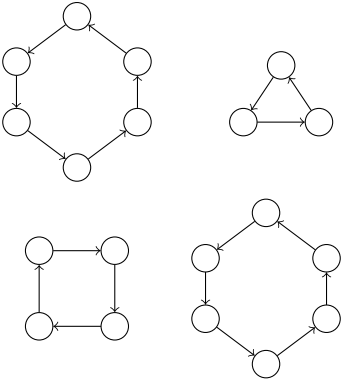
Min ATSP in polytime

- $O(\log n)$-approximation [FGM '82].
- $O(\frac{\log n}{\log \log n})$-approximation [AGMOS '10].

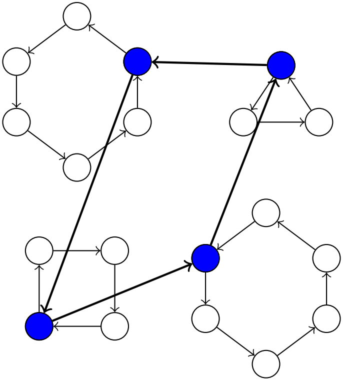
Our goal:

Theorem

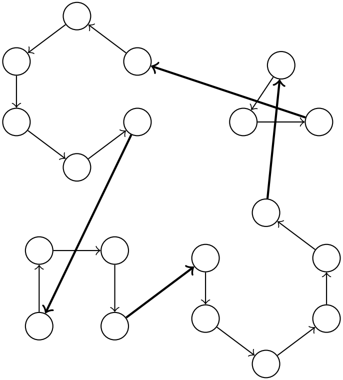
$\forall r \leqslant n$, *Min ATSP is* $\log r$*-approximable in time* $O^*(2^{n/r})$.



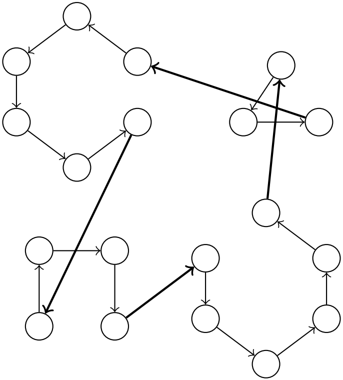
A circuit cover of minimum length can be found in polytime.



Pick any vertex in each cycle and recurse.



This can only decrease the total length (triangle inequality).



ratio = recursion depth: $\log n$ for polytime; $\log r$ for time $2^{n/r}$.

(Randomized) Exponential Time Hypothesis (ETH):

Assumption: no (randomized) $2^{o(n)}$-time algorithm solving 3-SAT.

Theorem (Sparsification Lemma, IPZ '01)

A $2^{o(n)}$-time algorithm for 3-SAT with $m \leqslant Cn$ disproves ETH.

Inapproximability in super-polynomial time

Theorem (CLN '13)

Under randomized ETH, $\forall \varepsilon > 0$, for all sufficiently big $r < n^{1/2-\varepsilon}$,

Max Independent Set is not r-approximable in time $2^{n^{1-\varepsilon}/r^{1+\varepsilon}}$.

Inapproximability in super-polynomial time

Theorem (CLN '13)

Under randomized ETH, $\forall \varepsilon > 0$, for all sufficiently big $r < n^{1/2-\varepsilon}$,

Max Independent Set is not r-approximable in time $2^{n^{1-\varepsilon}/r^{1+\varepsilon}}$.

SAT formula ϕ with N variables $\rightsquigarrow$ graph G with $r^{1+\varepsilon}N^{1+\varepsilon}$ vertices

- ϕ satisfiable $\Rightarrow \alpha(G) \approx rN^{1+\varepsilon}$.
- ϕ unsatisfiable $\Rightarrow \alpha(G) \approx r^{\varepsilon}N^{1+\varepsilon}$.

Inapproximability in super-polynomial time

Goal: Assuming ETH, Π is not r-approximable in time $2^{o(n/f(r))}$

Inapproximability in super-polynomial time

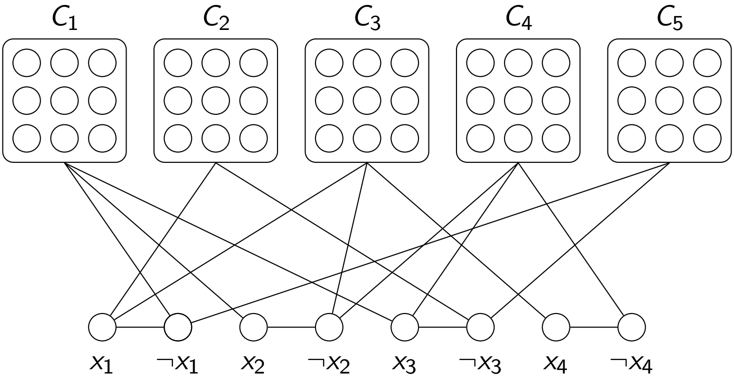
Goal: Assuming ETH, Π is not r-approximable in time $2^{o(n/f(r))}$

SAT formula ϕ with N variables $\rightsquigarrow \mathcal{I}$ instance of Π s.t.

- $|\mathcal{I}| \approx f(r)N$
- ϕ satisfiable $\Rightarrow$ val$(\Pi) \approx a$
- ϕ unsatisfiable $\Rightarrow$ val$(\Pi) \approx ra$

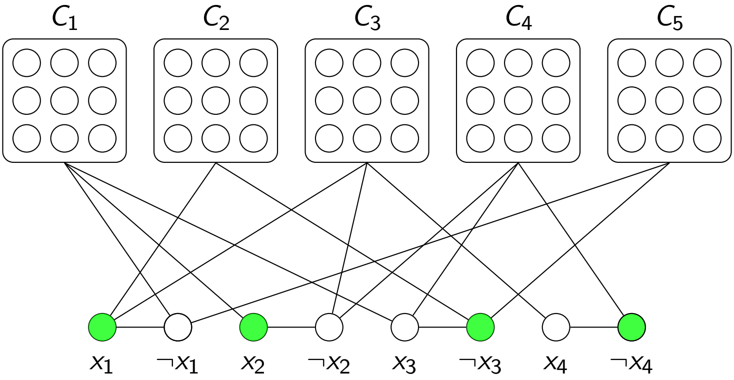
Min Independent Dominating Set

Inapproximability in polytime [I '91, H '93]



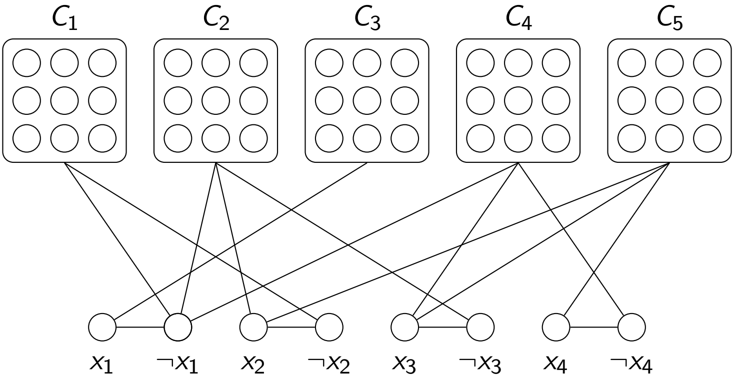
Satifiable CNF formula with N variables and CN clauses

Inapproximability in polytime [I '91, H '93]



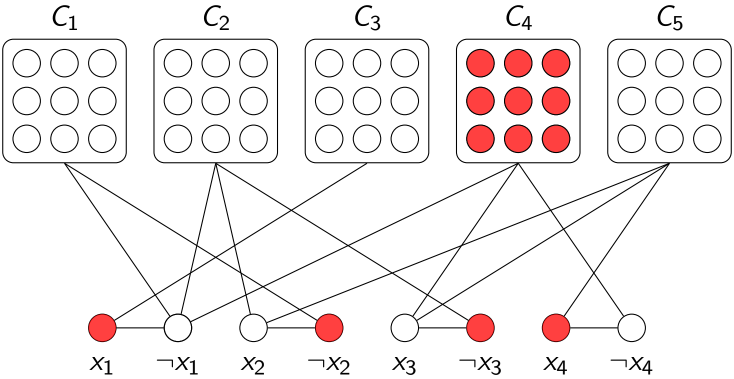
MIDS of size N

Inapproximability in polytime [I '91, H '93]



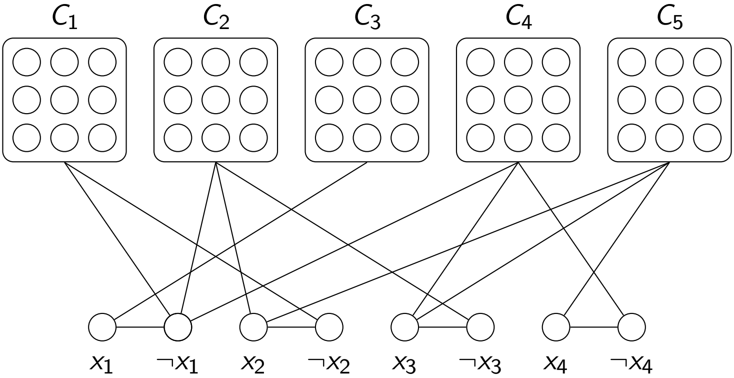
Unsatifiable CNF formula with N variables and CN clauses

Inapproximability in polytime [I '91, H '93]



MIDS of size greater than rN

Inapproximability in polytime [I '91, H '93]



Set $r = N^{9998} \approx n^{\frac{9998}{10000}} \geqslant n^{0.999}$
As $n = 2N + CrN^2 \approx N^{10000}$

(In)approximability in subexponential time

Our goal:

Theorem

Under ETH, $\forall \varepsilon > 0$, $\forall r \leqslant n$,

MIDS is not r-approximable in time $O^*(2^{n^{1-\varepsilon}/r^{1+\varepsilon}})$.

almost matching the r-approximation in time $O^*(2^{n \log(er)/r})$.

Inapproximability in subexponential time

Our goal:

Theorem

Under ETH, $\forall \varepsilon > 0$, $\forall r \leqslant n$,

MIDS is not r-approximable in time $O^*(2^{n^{1-\varepsilon}/r^{1+\varepsilon}})$.

In the previous reduction, $n \approx rN^2$.

We need to build a graph with $n \approx rN$ vertices.

Put only r vertices per independent set C_i and use the inapproximability of SAT to boost the gap.

Put only r vertices per independent set C_i and use the inapproximability of SAT to boost the gap.

- The number of vertices is now $n = 2N + rCN = O(rN)$.

Put only r vertices per independent set C_i and use the inapproximability of SAT to boost the gap.

- The number of vertices is now $n = 2N + rCN = O(rN)$.
- YES: MIDS of size N / NO: MIDS of size $> N + r\alpha CN$.

Put only r vertices per independent set C_i and use the inapproximability of SAT to boost the gap.

- The number of vertices is now $n = 2N + rCN = O(rN)$.
- YES: MIDS of size N / NO: MIDS of size $> N + r\alpha CN$.

There is an almost linear reduction from 3-SAT to 3-SAT introducing a constant gap [MR '08].

Max Minimal Vertex Cover

Approximability in polytime [BDP '13]

- MMVC admits a $n^{1/2}$-approximation,
- but no $n^{1/2-\varepsilon}$-approximation for any $\varepsilon > 0$, unless P=NP.

Approximability in polytime [BDP '13]

- MMVC admits a $n^{1/2}$-approximation,
- but no $n^{1/2-\varepsilon}$-approximation for any $\varepsilon > 0$, unless P=NP.

Our goal:

Theorem

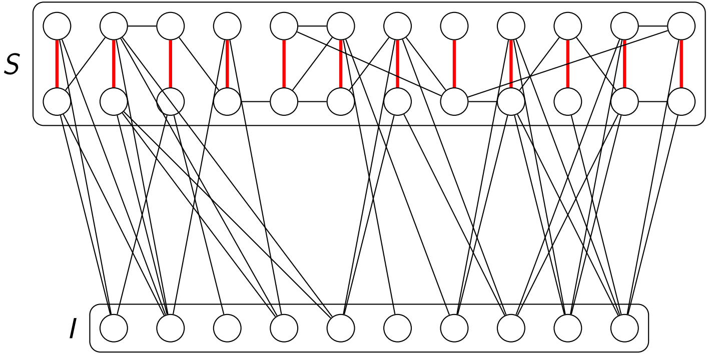
For any $r \leqslant n$, *MMVC is r-approximable in time $O^*(3^{n/r^2})$*.

Theorem

Under ETH, $\forall \varepsilon > 0$, $\forall r \leqslant n^{1/2-\varepsilon}$,

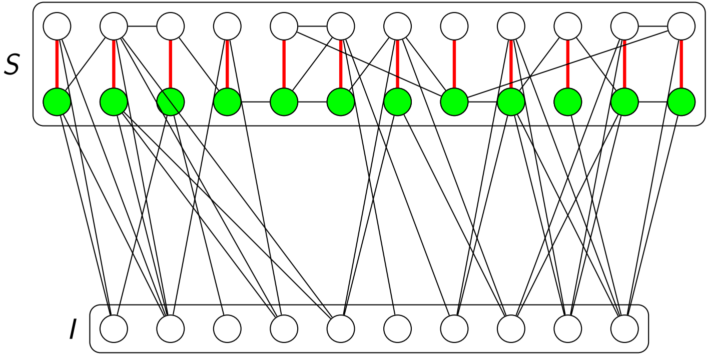
MMVC is not r-approximable in time $O^*(2^{n^{1-\varepsilon}/r^{2+\varepsilon}})$.

3-approximation bluff



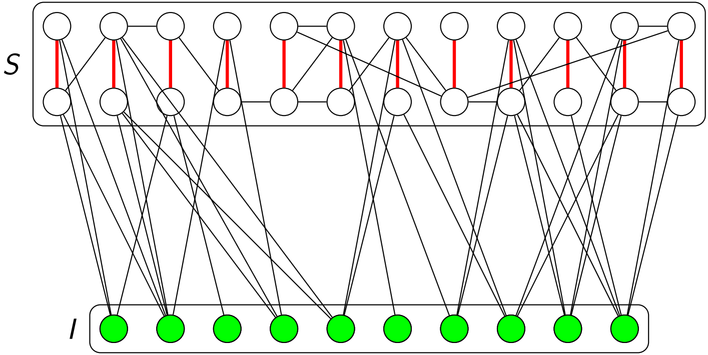
Let M be a maximal matching, $S := V(M)$, and $I := V \setminus S$.

3-approximation bluff



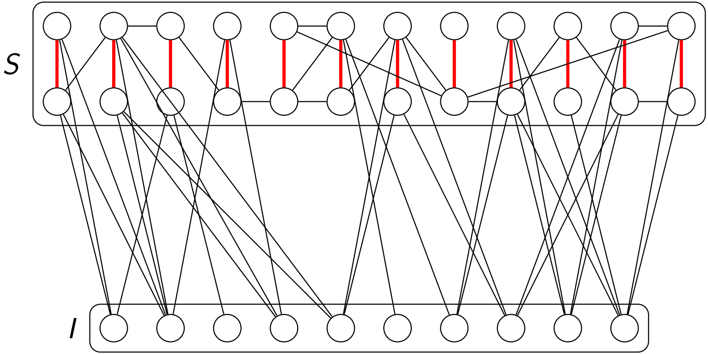
Each vertex of this half S' of S has at least one private edge.

3-approximation bluff



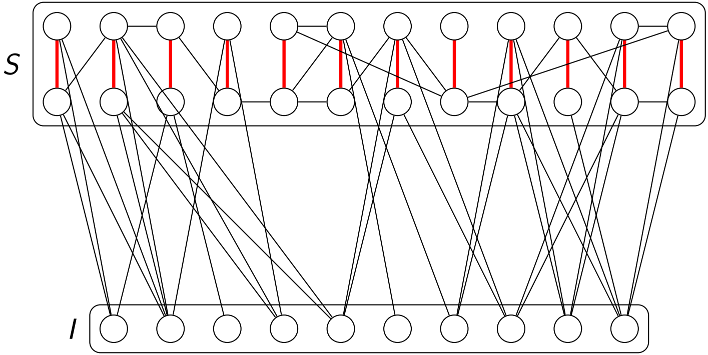
Each vertex of I has at least one private edge.

3-approximation bluff



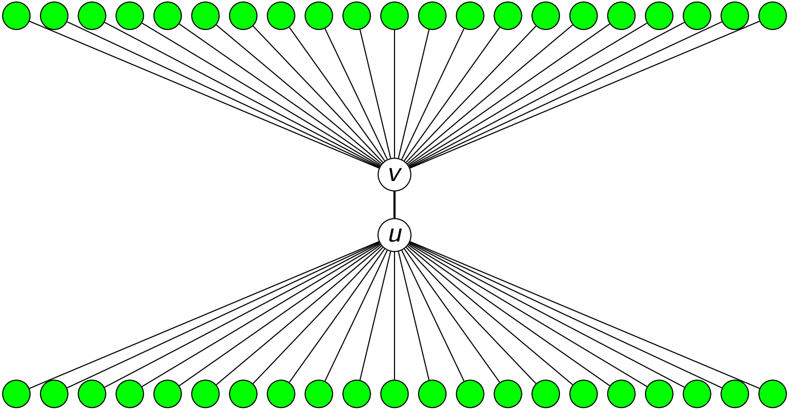
$S' \geqslant n/3$ or $I \geqslant n/3$.

3-approximation bluff



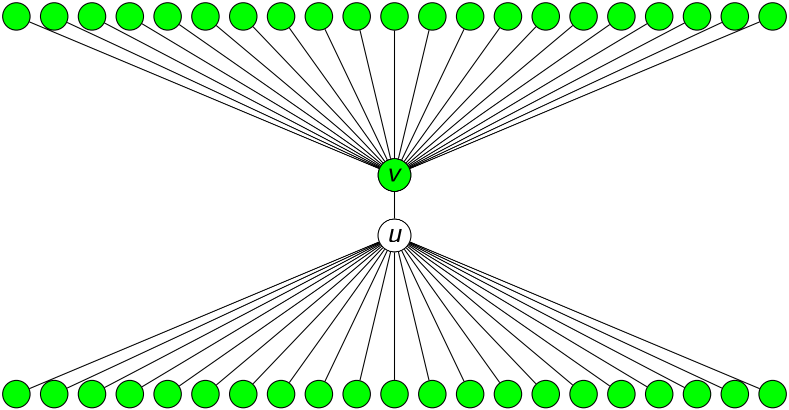
Completing S' and I into a vertex covers gives a 3-approximation.

How things actually work



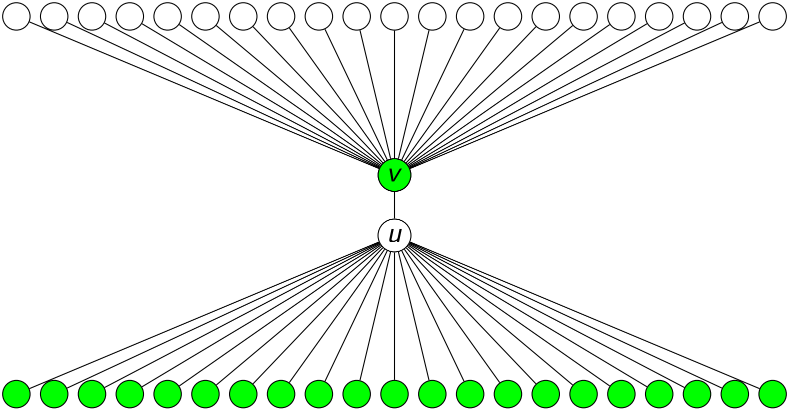
To cover uv, we should include u or v.

How things actually work



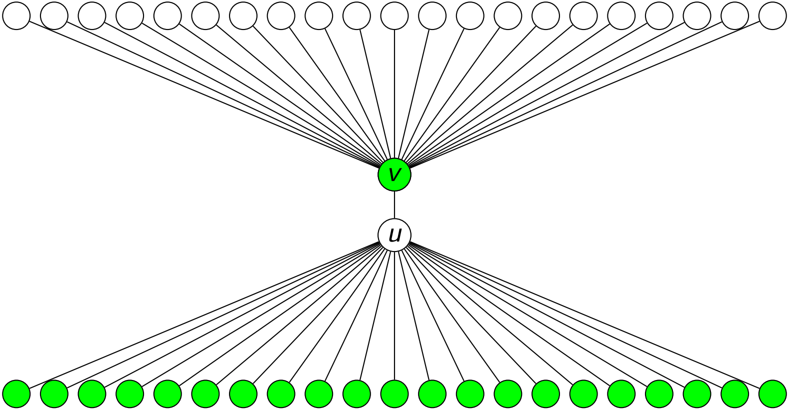
And then, we may *lose* many vertices.

How things actually work

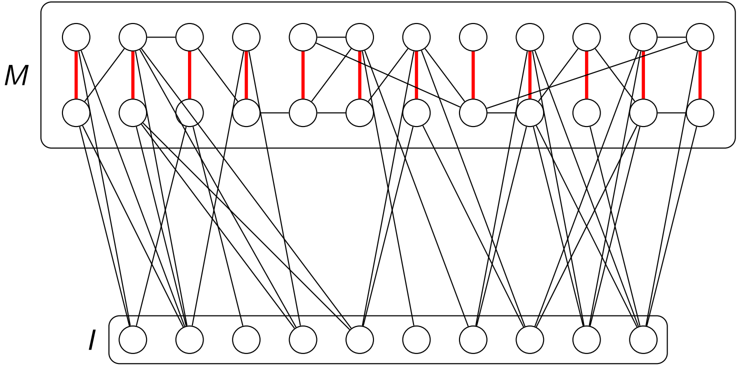


Removing vertices from vertex cover S, leads to a solution.

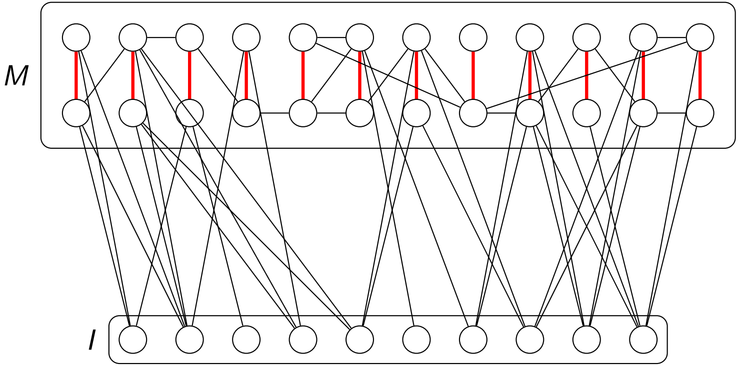
How things actually work



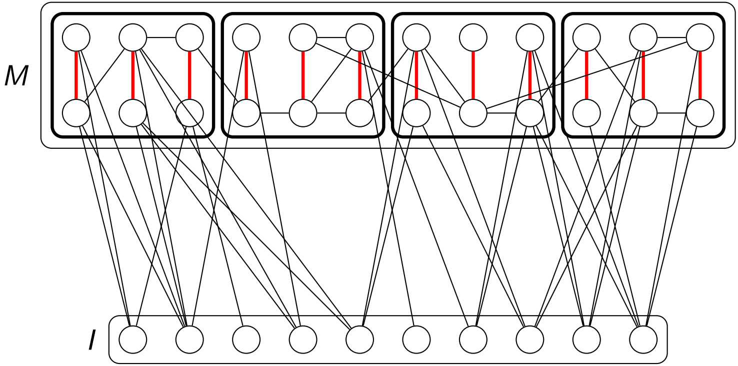
The solution is at least of size $N(\overline{S})$.



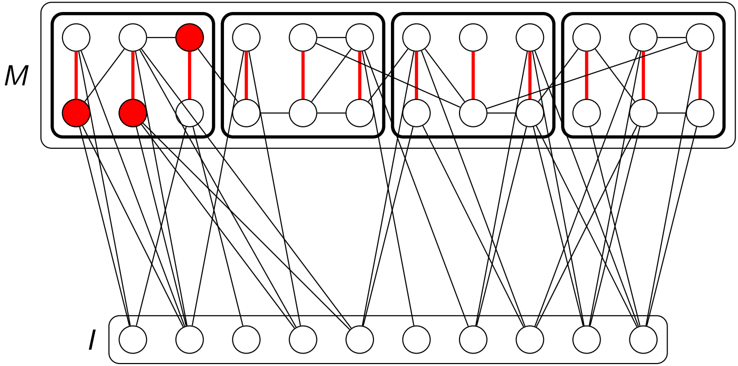
Compute any maximal matching M.



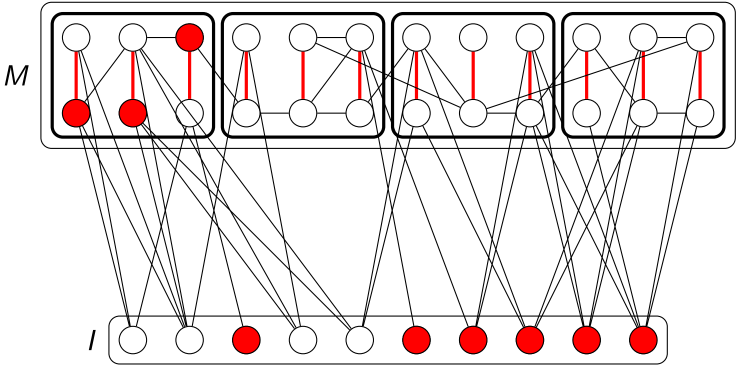
If $|M| \geqslant n/r$, then any (minimal) vertex cover contains $\geqslant n/r$.



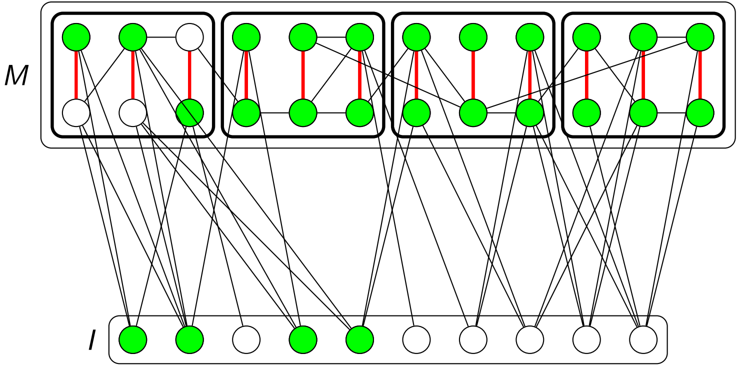
Otherwise split M into r parts $(A_1, A_2, \ldots, A_r)$ of size $\leqslant n/r^2$.



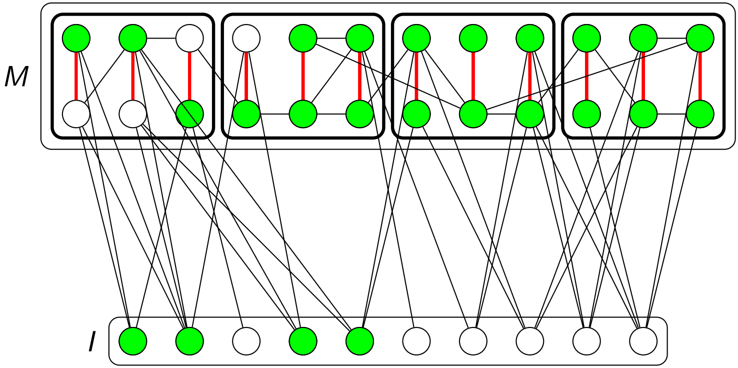
For each of the $\leqslant 3^{n/r^2}$ independent sets of each $G[A_i]$,



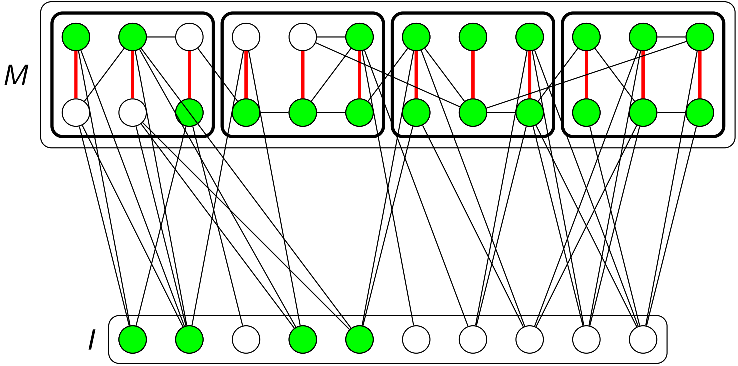
add all the non dominated vertices of I,



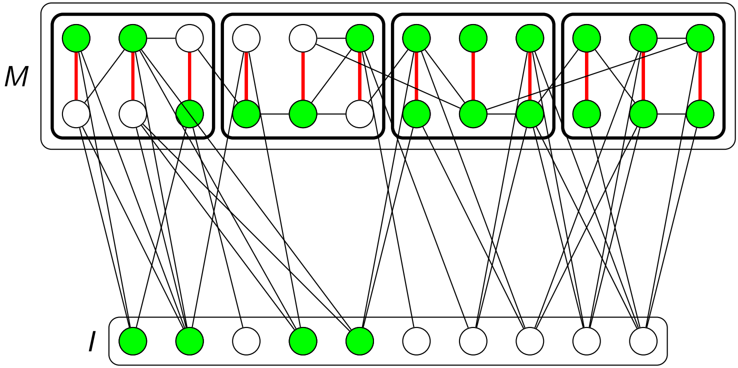
and compute a minimal vertex cover from the complement.



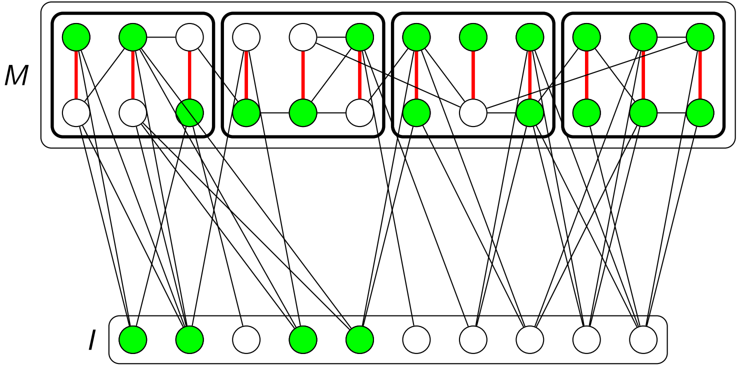
and compute a minimal vertex cover from the complement.



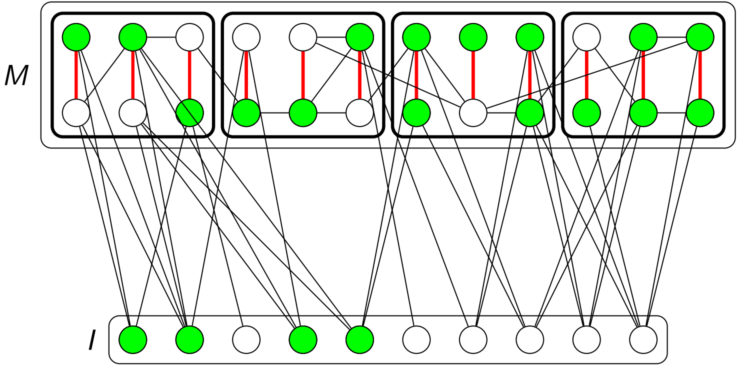
and compute a minimal vertex cover from the complement.



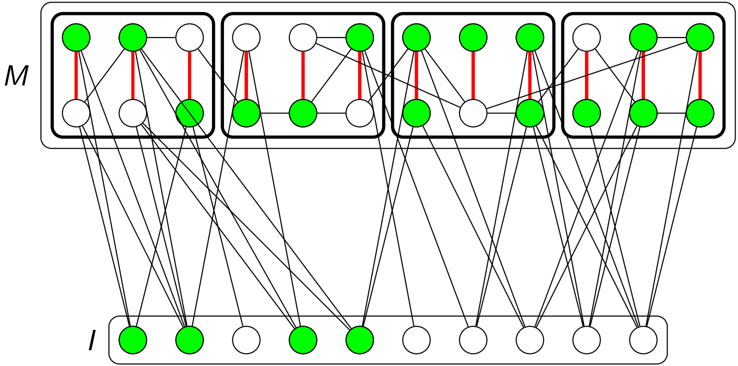
and compute a minimal vertex cover from the complement.



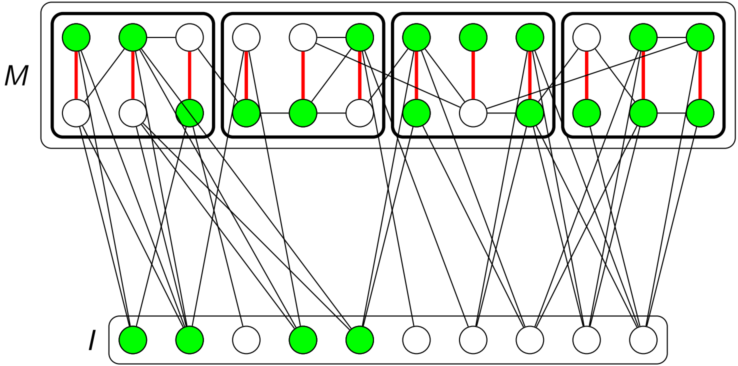
and compute a minimal vertex cover from the complement.



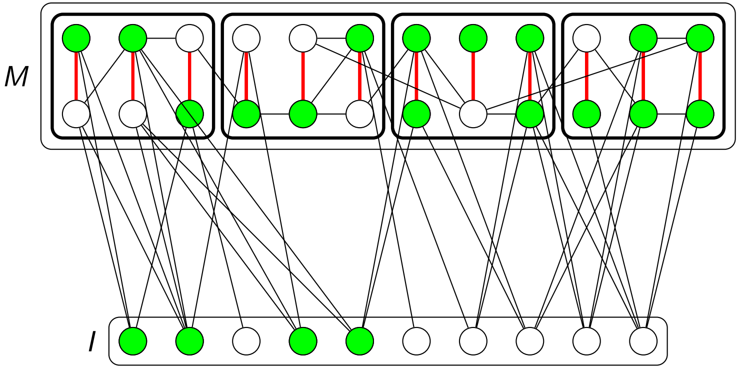
and compute a minimal vertex cover from the complement.



An optimal solution $R = N(\overline{R}) = N(\overline{R} \cap I) \cup \bigcup_i N(\overline{R} \cap A_i)$.

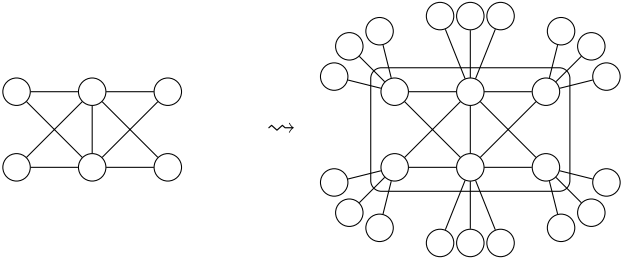


$$\exists i,\ |N(\overline{R} \cap I) \cup N(\overline{R} \cap A_i)| \geqslant \frac{|N(\overline{R})|}{r}.$$

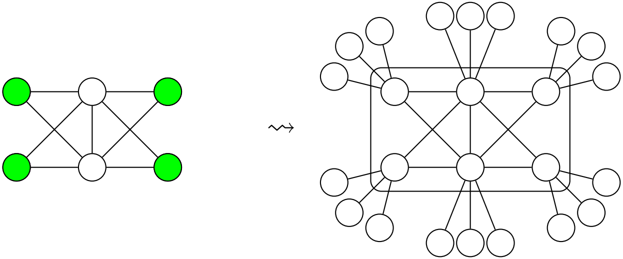


$\overline{R} \cap A_i$ will be tried, and completed with a superset of $\overline{R} \cap I$.

MIS ($\approx rN$ vertices) $\rightsquigarrow$ MMVC ($\approx r^2N$ vertices)

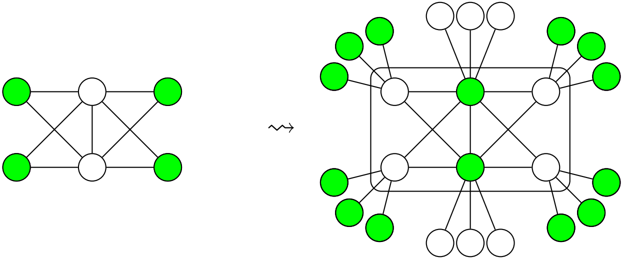


MIS ($\approx rN$ vertices) $\rightsquigarrow$ MMVC ($\approx r^2N$ vertices)



ϕ satisfiable $\Rightarrow |\text{IS}| \approx rN$; ϕ unsatisfiable $\Rightarrow |\text{IS}| \approx N$.

MIS ($\approx rN$ vertices) $\rightsquigarrow$ MMVC ($\approx r^2N$ vertices)



ϕ satisfiable $\Rightarrow |\text{MVC}| \approx r^2N$; ϕ unsatisfiable $\Rightarrow |\text{MVC}| \approx rN$.

Max Induced Path/Forest/Tree

Theorem

Under ETH, $\forall \varepsilon > 0$, $\forall r \leqslant n^{1/2-\varepsilon}$,

Max Induced Forest has no r-approximation in time $2^{n^{1-\varepsilon}/(2r)^{1+\varepsilon}}$.

A max induced forest has size in $[\alpha(G), 2\alpha(G)]$.

Theorem

Under ETH, $\forall \varepsilon > 0$, $\forall r \leqslant n^{1/2-\varepsilon}$,

Max Induced Forest has no r*-approximation in time* $2^{n^{1-\varepsilon}/(2r)^{1+\varepsilon}}$.

A max induced forest has size in $[\alpha(G), 2\alpha(G)]$.

- An independent set is a special forest.
- A forest has an independent set of size at least the half.

Theorem

Under ETH, $\forall \varepsilon > 0$, $\forall r \leqslant n^{1/2-\varepsilon}$,

Max Induced Tree has no r-approximation in time $2^{n^{1-\varepsilon}/(2r)^{1+\varepsilon}}$.

Add a universal vertex v to the gap instances of MIS: $G \rightsquigarrow G'$.

Theorem

Under ETH, $\forall \varepsilon > 0$, $\forall r \leqslant n^{1/2-\varepsilon}$,

Max Induced Tree has no r-approximation in time $2^{n^{1-\varepsilon}/(2r)^{1+\varepsilon}}$.

Add a universal vertex v to the gap instances of MIS: $G \rightsquigarrow G'$.

- G' has an induced tree of size $\alpha(G) + 1$.
- If T is an induced tree of G', $\alpha(G) \geqslant |T|/2$.

PCP-free inapproximability

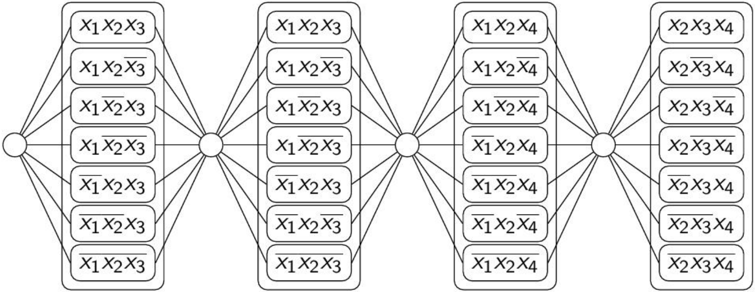
Our goal:

Theorem

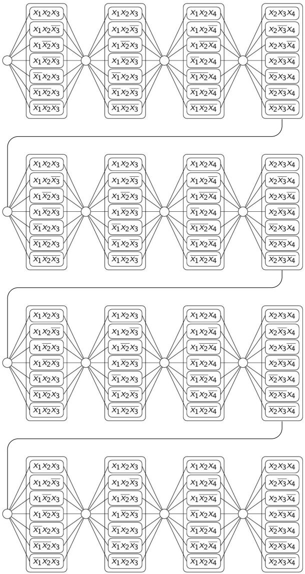
Under ETH, $\forall \varepsilon > 0$ and $\forall r \leqslant n^{1-\varepsilon}$,

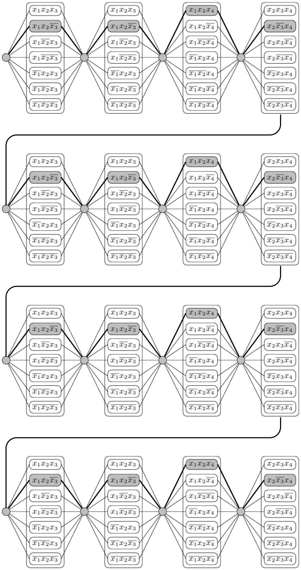
Max Induced Path has no r-approximation in time $2^{o(n/r)}$.

Walking through partial satisfying assignments



Contradicting edges are not represented





Open questions

- Is there an r-approximation in $O^*(2^{n/r})$ for MIDS? for Max Induced Matching?

Open questions

- Is there an r-approximation in $O^*(2^{n/r})$ for MIDS? for Max Induced Matching?
- Set Cover is $\log r$-approximable in time $O^*(2^{n/r})$ [CKW '09] but not in time $O^*(2^{(n/r)^\alpha})$ for some α [M' 11]. Can we tighten this lower bound?

Open questions

- Is there an r-approximation in $O^*(2^{n/r})$ for MIDS? for Max Induced Matching?
- Set Cover is $\log r$-approximable in time $O^*(2^{n/r})$ [CKW '09] but not in time $O^*(2^{(n/r)^\alpha})$ for some α [M' 11]. Can we tighten this lower bound?
- For Set Cover, we know a polytime $\sqrt{m}$-approximation [N '07] but only an r-approximation in time $O^*(2^{m/r})$ [CKW '09]. Can we match the upper and lower bounds?

Open questions

- Is there an r-approximation in $O^*(2^{n/r})$ for MIDS? for Max Induced Matching?
- Set Cover is $\log r$-approximable in time $O^*(2^{n/r})$ [CKW '09] but not in time $O^*(2^{(n/r)^\alpha})$ for some α [M' 11]. Can we tighten this lower bound?
- For Set Cover, we know a polytime $\sqrt{m}$-approximation [N '07] but only an r-approximation in time $O^*(2^{m/r})$ [CKW '09]. Can we match the upper and lower bounds?

Thank you for your attention!