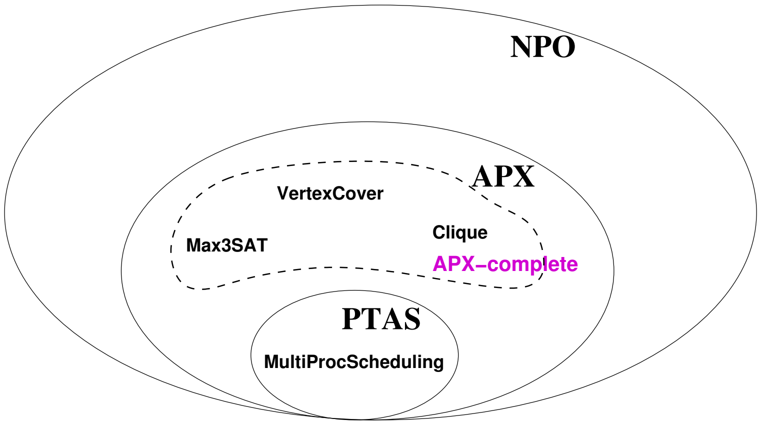
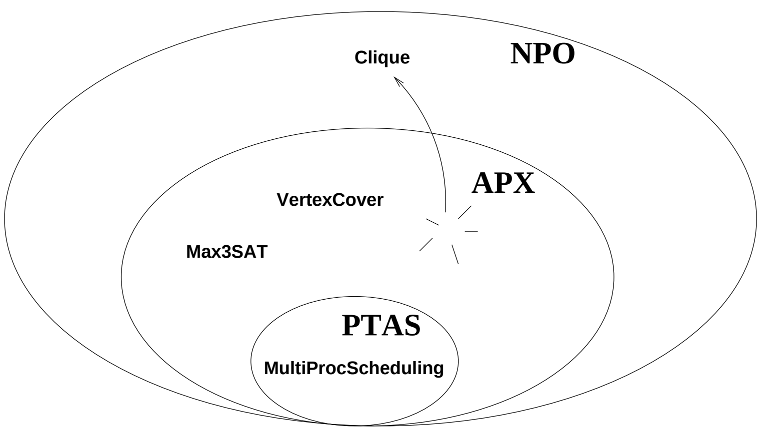
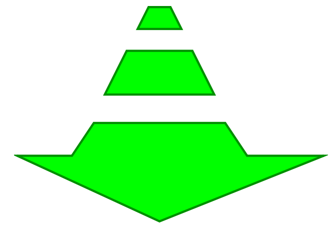




For any $\epsilon > 0$ approximating CLIQUE with ratio $1 + \epsilon$ is NP-hard, where n is the number of vertices in the input graph.



There is an $\epsilon > 0$ such that approximating CLIQUE with ratio n^{ϵ} is NP-hard, where n is the number of vertices in the input graph.

Significance

(Rather negative result) If ϵ was $\frac{1}{2}$ and the largest cliques in graphs on n vertices had size $O(\sqrt{n})$ we would not be guaranteed to find (in polynomial time) a clique with more than $O(1)$ vertices.

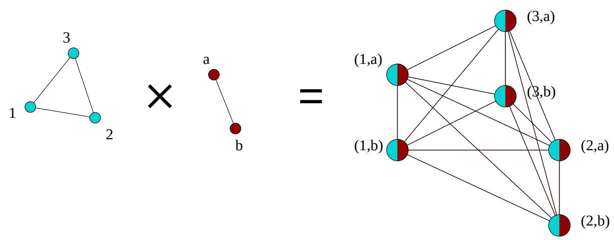
(Better picture in practice) On average the largest cliques of an n vertex graph chosen at random have size $O(\log n)$ (therefore exhaustive search should give us, in polynomial time, some good candidates).

Beyond APX, the CLIQUE problem

The hardness result for CLIQUE relies upon its interesting *self-improvement* behavior when we take graph products. The next example describes this behavior.

> For a graph $G = (V, E)$ let $\hat{E}$ denote E with all self-loops added, that is, $\hat{E} = E \cup \{(u, u) : u \in V\}$. For graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, their *product* $G_1 \times G_2$ is the graph whose vertex set is the set $V_1 \times V_2$, and edge set is
> $$\{((u_1, v_1), (u_2, v_2)) : (u_1, u_2) \in \hat{E}_1 \wedge (v_1, v_2) \in \hat{E}_2\}$$

Generalisation of G^k we saw last week.



Let $\omega(G)$ be the size of the largest clique in a graph. It is easily checked that $\omega(G_1 \times G_2) = \omega(G_1) \cdot \omega(G_2)$.

Self-improvement technique

Now suppose a reduction h exists from SAT to CLIQUE, such that the graph produced by the reduction has clique number either l, or $(1-\epsilon)l$, depending on whether or not the SAT formula was satisfiable or not.

Claim: It is NP-hard to approximate CLIQUE with any constant ratio.

Suppose we had a c-approximation algorithm A for CLIQUE.

Choose k so that $c < (1-\epsilon)^{-k}$ (e.g. $k = \lceil \log_{1/(1-\epsilon)} c \rceil$ would do).

To decide SAT reduce it to CLIQUE, then compute Let H, be the product of $h(G)$ with itself k times.

$\omega(H)$ is either l^k or $(1-\epsilon)^k l^k$.

But $c < (1-\epsilon)^{-k}$, hence you can use A to verify whether the optimal cliques in $h(G)^k$ are "small" or "large".

This decides SAT deterministically in polynomial time!

Key property. (Self-improvement) The gap in clique numbers, $(1-\epsilon)^{-k}$, can be made arbitrarily large by increasing k enough.

Note however that H has size $n^{O(k)}$, so k must remain $O(1)$ if the above construction has to work in polynomial time.

The rapid increase in problem size when using self-improvement may seem hard to avoid. Surprisingly, the following combinatorial object often allows us to do just that.

> Let n be an integer. A (n, k, α) *booster* is a collection $\mathcal{S}$ of subsets of $\{1, 2, \ldots, n\}$, each of size k. For every subset $A \subseteq \{1, 2, \ldots, n\}$, the sets in the collection that are subsets of A constitute a fraction[a] between $(\frac{|A|}{n} - \alpha)^k$ and $(\frac{|A|}{n} + \alpha)^k$ of all sets in $\mathcal{S}$.

[a] When $\frac{|A|}{n} < 1.1\alpha$, the quantity $(\frac{|A|}{n} - \alpha)^k$ should be considered to be 0.

Example

The set $\mathcal{P}(n)$ of all subsets of $\{1, 2, \ldots, n\}$ of size k is a booster with $\alpha \sim 0$. For any $A \subseteq \{1, 2, \ldots, n\}$, with $|A| = O(n)$, the fraction of subsets of A in $\mathcal{P}(n)$ is $\binom{|A|}{k} / \binom{n}{k}$, which is approximately $(|A|/n)^k$.

Unfortunately $|\mathcal{P}(n)| = \binom{n}{k} = O(n^k)$, hence k must be $O(1)$ if the booster has to be used in polynomial-time reductions.

In the following treatment we would like k to be as large as possible.

More specifically, take $n = 7$ and $k = 3$. The following collection of subsets of $\{1, 2, 3, 4, 5, 6, 7\}$ is a (3,7,0)-booster.

$$\begin{array}{lllll}
\{1,2,3\}, & \{1,2,4\}, & \{1,3,4\}, & \{2,3,4\}, & \{1,2,5\}, \\
\{1,3,5\}, & \{2,3,5\}, & \{1,4,5\}, & \{2,4,5\}, & \{3,4,5\}, \\
\{1,2,6\}, & \{1,3,6\}, & \{2,3,6\}, & \{1,4,6\}, & \{2,4,6\}, \\
\{3,4,6\}, & \{1,5,6\}, & \{2,5,6\}, & \{3,5,6\}, & \{4,5,6\}, \\
\{1,2,7\}, & \{1,3,7\}, & \{2,3,7\}, & \{1,4,7\}, & \{2,4,7\}, \\
\{3,4,7\}, & \{1,5,7\}, & \{2,5,7\}, & \{3,5,7\}, & \{4,5,7\}, \\
\{1,6,7\}, & \{2,6,7\}, & \{3,6,7\}, & \{4,6,7\}, & \{5,6,7\}
\end{array}$$

Take set $A = \{1, 2, 5, 6, 7\}$. There are $\binom{5}{3} = 10$ elements of the booster that are subsets of A:

$$\begin{array}{lllll}
\{1,2,5\}, & \{1,2,6\}, & \{1,5,6\}, & \{2,5,6\}, & \{1,2,7\}, \\
\{1,5,7\}, & \{2,5,7\}, & \{1,6,7\}, & \{2,6,7\}, & \{5,6,7\}
\end{array}$$

and $0.2857 = \frac{10}{35} \sim \left(\frac{|A|}{n}\right)^3 = 0.3644$.

> *Technical result*
> For any $k = O(\log n)$ and $\alpha > 0$, an (n, k, α) booster of size poly(n) can be constructed in poly(n) time.

The proof of this result is beyond the scope of this module, but it has very important consequences in our main argument.

Booster product

Let G be a graph on n vertices. The *booster product* of G, $\mathcal{B}(G, n, k, \alpha)$ is a graph whose vertices are the sets of a (n, k, α)-booster $\mathcal{S}$, and there is an edge between sets S_i and S_j if and only if $S_i \cup S_j$ is a clique in G.

What is this? Why do we need all this?

> For any graph G, and any (n, k, α) booster, the clique number of the booster product of G lies between $(\frac{\omega(G)}{n} - \alpha)^k |\mathcal{S}|$ and $(\frac{\omega(G)}{n} + \alpha)^k |\mathcal{S}|$.

In other words:

Option 1 Using graph products we go from $\omega(G)$ to $\omega(G)^k$.

Option 2 Using booster products we go from $\omega(G)$ to $\omega(G)^k n^{O(1)}$

Using option 2 we "inflate" the gap even more!

Argument

Let $A \subseteq \{1, 2, \ldots, n\}$ be a clique of size $\omega(G)$ in graph G.

Then the number of sets from $\mathcal{S}$ that are subsets of A is between $(\frac{\omega(G)}{n} - \alpha)^k |\mathcal{S}|$ and $(\frac{\omega(G)}{n} + \alpha)^k |\mathcal{S}|$.

Clearly, all such sets form a clique in the booster product.

Conversely, given the largest clique B in the booster product, let A be the union of all sets in the clique.

Then A is a clique in G, and hence must have size at most $\omega(G)$.

The booster property implies that the size of B is as claimed.

Finally we come to the most important consequence of the argument so far:

> There is an $\epsilon > 0$ such that approximating CLIQUE with ratio n^ϵ is NP-hard, where n is the number of vertices in the input graph.

(1) Let G be the graph obtained from the usual 3SAT to CLIQUE reduction, and suppose it has N vertices (remember that N is three times the number of clauses in the original formula).

(2) The reduction ensures, for some fixed $\beta > 0$, that $\omega(G)$ is either at least $N/3$ or at most $N(1-\beta)/3$, and it is NP-hard to decide which case holds.

(3) Now construct a $(N, \log N, \alpha)$ booster, $\mathcal{S}$, by choosing $\alpha = \beta/9$.

(4) Construct the booster product of G. The number of vertices in the booster product is $|\mathcal{S}|$, and Lemma above says the clique number is either at least $((3-\beta)/9)^{\log N}|\mathcal{S}|$ or at most $((3-2\beta)/9)^{\log N}|\mathcal{S}|$. Hence the gap is now N^{γ} for some $\gamma > 0$, and further, $|\mathcal{S}| = N^{O(1)}$, so this gap is $|\mathcal{S}|^{\epsilon}$ for some $\epsilon > 0$.

That's all folks!

- Two major design paradigms: greedy + dynamic programming.
- A number of application areas. We looked more deeply into string algorithms and graph matching algorithms.
- A quick glimpse in space complexity theory.
- The broad and very widely used area approximation algorithms.

What should you keep of this?

- The definition of greedy + dynamic programming algorithms, and the “feeling” that these paradigms can lead to very good quality solutions in some cases.
- Advises on how to design a program in several different cases.
- The knowledge of what an approximation algorithm is and how can you prove results about them.

  Many big industrial projects in Europe (e.g. a recent railway optimisation project in Switzerland, Germany, Italy, Greece).