KNAPSACK Has an Approximation Threshold of Zero[a]

Theorem 85 *For any ϵ, there is a polynomial-time ϵ-approximation algorithm for* KNAPSACK.

- We have n weights $w_1, w_2, \ldots, w_n \in \mathbb{Z}^+$, a weight limit W, and n values $v_1, v_2, \ldots, v_n \in \mathbb{Z}^+$.[b]
- We must find an $I \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i$ is the largest possible.

---

[a] Ibarra & Kim (1975). This algorithm can be used to derive good approximation algorithms for some NP-complete scheduling problems (Bansal & Sviridenko, 2006).

[b] If the values are fractional, the result is slightly messier, but the main conclusion remains correct. Contributed by Mr. Jr-Ben Tian (`B89902011`, `R93922045`) on December 29, 2004.

The Proof (continued)

- Let
$$V = \max\{ v_1, v_2, \ldots, v_n \}.$$
- Clearly, $\sum_{i \in I} v_i \leq nV$.
- Let $0 \leq i \leq n$ and $0 \leq v \leq nV$.
- $W(i, v)$ is the minimum weight attainable by selecting only from the *first* i items and with a total value of v.
  - It is an $(n + 1) \times (nV + 1)$ table.

The Proof (continued)

- Set $W(0, v) = \infty$ for $v \in \{1, 2, \ldots, nV\}$ and $W(i, 0) = 0$ for $i = 0, 1, \ldots, n$.[a]
- Then, for $0 \leq i < n$ and $1 \leq v \leq nV$,[b]

$$W(i+1, v) = \begin{cases} \min\{ W(i, v), W(i, v - v_{i+1}) + w_{i+1} \}, & \text{if } v_{i+1} \leq v, \\ W(i, v), & \text{otherwise.} \end{cases}$$

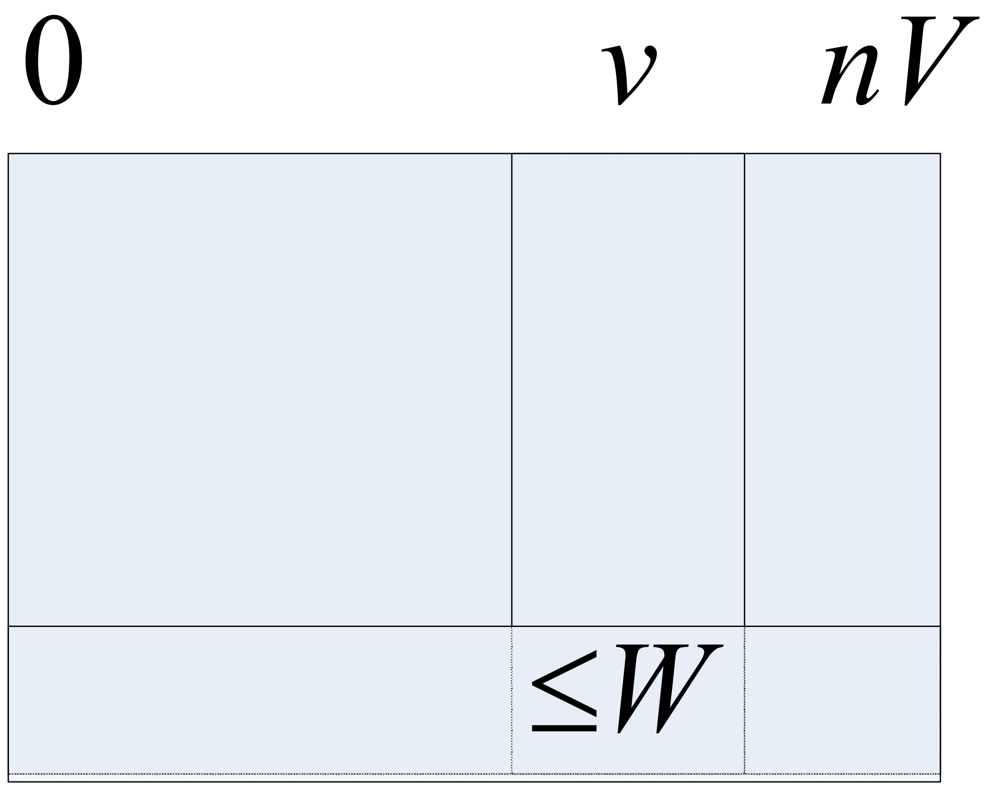
- Finally, pick the largest v such that $W(n, v) \leq W$.[c]

---

[a] Contributed by Mr. Ren-Shuo Liu (`D98922016`) and Mr. Yen-Wei Wu (`D98922013`) on December 28, 2009.

[b] The textbook's formula has an error here.

[c] Lawler (1979).


0
v
nV
≤W

The Proof (continued)

With 6 items, values $(4, 3, 3, 3, 2, 3)$, weights $(3, 3, 1, 3, 2, 1)$, and $W = 12$, the maximum total value 16 is achieved with $I = \{1, 2, 3, 4, 6\}$; I's weight is 11.

0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	∞	∞	∞	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	∞	∞	3	3	∞	∞	6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	∞	∞	1	3	∞	4	4	∞	∞	7	∞	∞	∞	∞	∞	∞	∞	∞
0	∞	∞	1	3	∞	4	4	∞	7	7	∞	∞	10	∞	∞	∞	∞	∞
0	∞	2	1	3	3	4	4	6	6	7	9	9	10	∞	12	∞	∞	∞
0	∞	2	1	3	3	2	4	4	5	5	7	7	8	10	10	11	∞	13

The Proof (continued)

- The running time $O(n^2 V)$ is not polynomial.
- Call the problem instance

$$x = (w_1, \ldots, w_n, W, v_1, \ldots, v_n).$$

- Additional idea: Limit the number of precision bits.
- Define

$$v_i' = \left\lfloor \frac{v_i}{2^b} \right\rfloor.$$

- Note that

$$v_i - 2^b < 2^b v_i' \leq v_i. \tag{23}$$

The Proof (continued)

- Call the approximate instance

$$x' = (w_1, \ldots, w_n, W, v'_1, \ldots, v'_n).$$

- Solving x' takes time $O(n^2 V/2^b)$.
  - Use $v'_i = \lfloor v_i/2^b \rfloor$ and $V' = \max(v'_1, v'_2, \ldots, v'_n)$ in the dynamic programming.
  - It is now an $(n+1) \times (nV+1)/2^b$ table.
- The selection I' is optimal for x'.
- But I' may not be optimal for x, although it still satisfies the weight budget W.

The Proof (continued)

With the same parameters as p. 782 and $b = 1$: Values are $(2, 1, 1, 1, 1, 1)$ and the optimal selection $I' = \{1, 2, 3, 5, 6\}$ for x' has a *smaller* maximum value $4 + 3 + 3 + 2 + 3 = 15$ for x than I's 16; its weight is $10 < W = 12$.[a]

0	∞	∞	∞	∞	∞	∞	∞
0	∞	3	∞	∞	∞	∞	∞
0	3	3	6	∞	∞	∞	∞
0	1	3	4	7	∞	∞	∞
0	1	3	4	7	10	∞	∞
0	1	3	4	6	9	12	∞
0	1	2	4	5	7	10	13

[a]The *original* optimal $I = \{1, 2, 3, 4, 6\}$ on p. 782 has the same value 6 and but higher weight 11 for x'.

The Proof (continued)

- The value of I' for x is close to that of the optimal I as

$$
\begin{aligned}
& \sum_{i \in I'} v_i \\
\geq \;& \sum_{i \in I'} 2^b v_i' \quad \text{by inequalities (23) on p. 783} \\
= \;& 2^b \sum_{i \in I'} v_i' \geq 2^b \sum_{i \in I} v_i' = \sum_{i \in I} 2^b v_i' \\
\geq \;& \sum_{i \in I} \left(v_i - 2^b \right) \quad \text{by inequalities (23)} \\
\geq \;& \left(\sum_{i \in I} v_i \right) - n 2^b.
\end{aligned}
$$

The Proof (continued)

- In summary,

$$\sum_{i \in I'} v_i \geq \left(\sum_{i \in I} v_i \right) - n2^b.$$

- Without loss of generality, assume $w_i \leq W$ for all i.
  - Otherwise, item i is redundant and can be removed early on.
- V is a lower bound on OPT.
  - Picking one single item with value V is a legitimate choice.

The Proof (concluded)

- The relative error from the optimum is:

$$\frac{\sum_{i \in I} v_i - \sum_{i \in I'} v_i}{\sum_{i \in I} v_i} \leq \frac{n 2^b}{V}.$$

- Suppose we pick $b = \lfloor \log_2 \frac{\epsilon V}{n} \rfloor$.
- The algorithm becomes ϵ-approximate.[a]
- The running time is then $O(n^2 V / 2^b) = O(n^3/\epsilon)$, a polynomial in n and $1/\epsilon$.[b]

---

[a]See Eq. (18) on p. 734.

[b]It hence depends on the *value* of $1/\epsilon$. Thanks to a lively class discussion on December 20, 2006. If we fix ϵ and let the problem size increase, then the complexity is cubic. Contributed by Mr. Ren-Shan Luoh (D97922014) on December 23, 2008.

Comments

- INDEPENDENT SET and NODE COVER are reducible to each other (Corollary 46, p. 382).
- NODE COVER has an approximation threshold at most 0.5 (p. 747).
- But INDEPENDENT SET is unapproximable (see the textbook).
- INDEPENDENT SET limited to graphs with degree $\leq k$ is called k-DEGREE INDEPENDENT SET.
- k-DEGREE INDEPENDENT SET is approximable (see the textbook).

On P vs. NP

If 50 million people believe a foolish thing,
it's still a foolish thing.
— George Bernard Shaw (1856–1950)

Exponential Circuit Complexity for NP-Complete Problems

- We shall prove exponential lower bounds for NP-complete problems using *monotone* circuits.
  - Monotone circuits are circuits without $\neg$ gates.[a]
- Note that this result does *not* settle the P vs. NP problem.

---

[a] Recall p. 320.

The Power of Monotone Circuits

- Monotone circuits can only compute monotone boolean functions.
- They are powerful enough to solve a P-complete problem: MONOTONE CIRCUIT VALUE (p. 321).
- There are NP-complete problems that are not monotone; they cannot be computed by monotone circuits at all.
- There are NP-complete problems that are monotone; they can be computed by monotone circuits.
  - HAMILTONIAN PATH and CLIQUE.

$\text{CLIQUE}_{n,k}$

- $\text{CLIQUE}_{n,k}$ is the boolean function deciding whether a graph $G = (V, E)$ with n nodes has a clique of size k.
- The input gates are the $\binom{n}{2}$ entries of the adjacency matrix of G.
  - Gate g_{ij} is set to true if the associated undirected edge $\{ i, j \}$ exists.
- $\text{CLIQUE}_{n,k}$ is a monotone function.
- Thus it can be computed by a monotone circuit.
- This does not rule out that *non*monotone circuits for $\text{CLIQUE}_{n,k}$ may use *fewer* gates.

Crude Circuits

- One possible circuit for $\text{CLIQUE}_{n,k}$ does the following.
  1. For each $S \subseteq V$ with $|S| = k$, there is a circuit with $O(k^2)$ $\wedge$-gates testing whether S forms a clique.
  2. We then take an OR of the outcomes of all the $\binom{n}{k}$ subsets $S_1, S_2, \ldots, S_{\binom{n}{k}}$.
- This is a monotone circuit with $O(k^2 \binom{n}{k})$ gates, which is exponentially large unless k or $n - k$ is a constant.
- A **crude circuit** $\text{CC}(X_1, X_2, \ldots, X_m)$ tests if there is an $X_i \subseteq V$ that forms a clique.
  - The above-mentioned circuit is $\text{CC}(S_1, S_2, \ldots, S_{\binom{n}{k}})$.

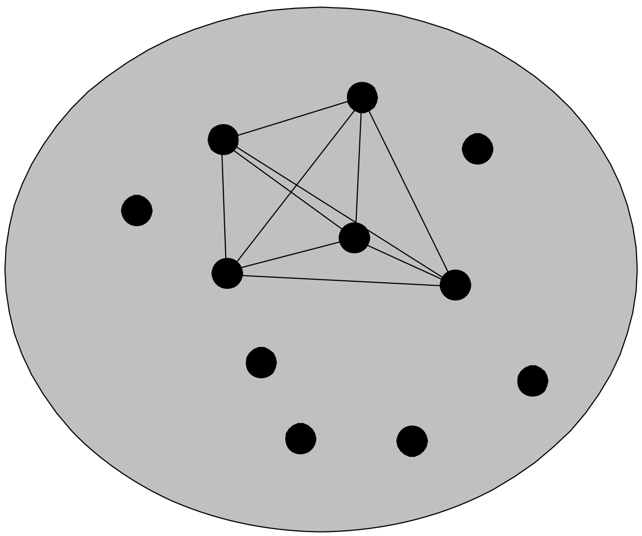
The Proof: Positive Examples

- Analysis will be applied to only the following **positive examples** and **negative examples** as input graphs.
- A positive example is a graph that has $\binom{k}{2}$ edges connecting k nodes in all possible ways.
- There are $\binom{n}{k}$ such graphs.
- They all should elicit a true output from $\text{CLIQUE}_{n,k}$.

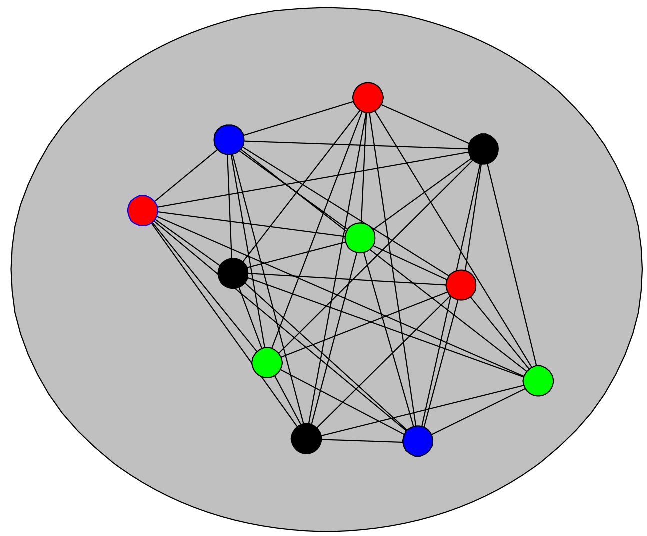
The Proof: Negative Examples

- Color the nodes with $k-1$ different colors and join by an edge any two nodes that are colored differently.
- There are $(k-1)^n$ such graphs.
- They all should elicit a false output from $\textsc{clique}_{n,k}$.
  - Each set of k nodes must have 2 identically colored nodes; hence there is no edge between them.

Positive and Negative Examples with $k = 5$



A positive example



A negative example

A Warmup to Razborov's (1985) Theorem[a]

Lemma 86 (The birthday problem) *The probability of collision, $C(N, q)$, when q balls are thrown randomly into $N \geq q$ bins is at most*

$$\frac{q(q-1)}{2N}.$$

Lemma 87 *If crude circuit $CC(X_1, X_2, \ldots, X_m)$ computes $\text{CLIQUE}_{n,k}$, then $m \geq n^{n^{1/8}/20}$ for n sufficiently large.*

[a] Arora & Barak (2009).

The Proof (continued)

- Let $k = n^{1/4}$.
- Let $\ell = \sqrt{k}/10$.
- Let $X \subseteq V$.

The Proof (continued)

- Suppose $|X| \leq \ell$.
- A random $f : X \to \{1, 2, \ldots, k-1\}$ has collisions with probability less than 0.01 by Lemma 86 (p. 799).
- Hence f is one-to-one with probability 0.99.
- When f is one-to-one, f is a coloring of X with $k-1$ colors without repeated colors.
- As a result, when f is one-to-one, it generates a clique on X.

The Proof (continued)

- Note that a random negative example is simply a random $g : V \to \{1, 2, \ldots, k-1\}$.
- So our random $f : X \to \{1, 2, \ldots, k-1\}$ is simply a random g restricted to X.
- In summary, the probability that X is not a clique when supplied with a random negative example is at most 0.01.

The Proof (continued)

- Now suppose $|X| > \ell$.
- Consider the probability that X is a clique when supplied with a random positive example.
- It is the probability that X is part of the clique.
- Hence the desired probability is at most

$$\frac{\binom{n-\ell}{k-\ell}}{\binom{n}{k}}.$$

The Proof (continued)

- Now,

$$\begin{aligned}
\frac{\binom{n-\ell}{k-\ell}}{\binom{n}{k}} &= \frac{k(k-1)\cdots(k-\ell+1)}{n(n-1)\cdots(n-\ell+1)} \\
&\leq \left(\frac{k}{n}\right)^{\ell} \\
&\leq n^{-(3/4)\,\ell} \\
&\leq n^{-\sqrt{k}/20} \\
&= n^{-n^{1/8}/20}.
\end{aligned}$$

The Proof (concluded)

- In summary, the probability that X is a clique when supplied with a random positive example is at most

$$n^{-n^{1/8}/20}.$$

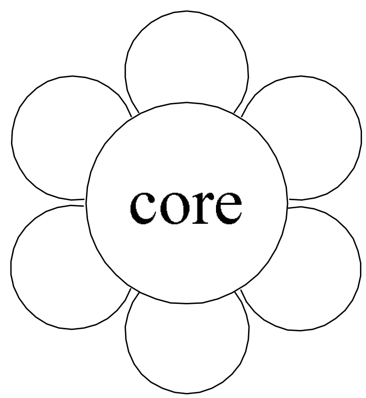
- So we need at least

$$n^{n^{1/8}/20}$$

  Xs in the crude circuit.

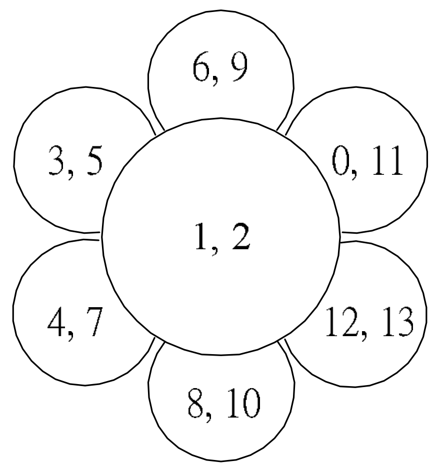
Sunflowers

- Fix $p \in \mathbb{Z}^+$ and $\ell \in \mathbb{Z}^+$.
- A **sunflower** is a family of p sets $\{ P_1, P_2, \ldots, P_p \}$, called **petals**, each of cardinality at most ℓ.
- Furthermore, all pairs of sets in the family must have the same intersection (called the **core** of the sunflower).



A Sample Sunflower

$$\{\{1,2,3,5\},\{1,2,6,9\},\{0,1,2,11\},$$
$$\{1,2,12,13\},\{1,2,8,10\},\{1,2,4,7\}\}.$$



The Erdős-Rado Lemma

Lemma 88 *Let $\mathcal{Z}$ be a family of more than $M \stackrel{\Delta}{=} (p-1)^\ell \ell!$ nonempty sets, each of cardinality ℓ or less. Then $\mathcal{Z}$ must contain a sunflower (with p petals).*

- Induction on ℓ.
- For $\ell = 1$, p different singletons form a sunflower (with an empty core).
- Suppose $\ell > 1$.
- Consider a *maximal* subset $\mathcal{D} \subseteq \mathcal{Z}$ of *disjoint* sets.
  - Every set in $\mathcal{Z} - \mathcal{D}$ intersects some set in $\mathcal{D}$.

The Proof of the Erdős-Rado Lemma (continued)

For example,

$$
\begin{aligned}
\mathcal{Z} &= \{\{1,2,3,5\},\{1,3,6,9\},\{0,4,8,11\},\\
&\quad \{4,5,6,7\},\{5,8,9,10\},\{6,7,9,11\}\},\\
\mathcal{D} &= \{\{1,2,3,5\},\{0,4,8,11\}\}.
\end{aligned}
$$

The Proof of the Erdős-Rado Lemma (continued)

- Suppose $\mathcal{D}$ contains at least p sets.
  - $\mathcal{D}$ constitutes a sunflower with an empty core.
- Suppose $\mathcal{D}$ contains fewer than p sets.
  - Let C be the union of all sets in $\mathcal{D}$.
  - $|\, C \,| \leq (p-1)\ell$.
  - C intersects every set in $\mathcal{Z}$ by $\mathcal{D}$'s maximality.
  - There is a $d \in C$ that intersects more than $\frac{M}{(p-1)\ell} = (p-1)^{\ell-1}(\ell-1)!$ sets in $\mathcal{Z}$.
  - Consider $\mathcal{Z}' = \{\, Z - \{\, d \,\} : Z \in \mathcal{Z}, d \in Z \,\}$.

The Proof of the Erdős-Rado Lemma (concluded)

- (continued)
  - $\mathcal{Z}'$ has more than $M' \stackrel{\Delta}{=} (p-1)^{\ell-1}(\ell-1)!$ sets.
  - M' is just M with ℓ replaced with $\ell - 1$.
  - $\mathcal{Z}'$ contains a sunflower by induction, say
  $$\{ P_1, P_2, \ldots, P_p \}.$$
  - Now,
  $$\{ P_1 \cup \{ d \}, P_2 \cup \{ d \}, \ldots, P_p \cup \{ d \} \}$$
  is a sunflower in $\mathcal{Z}$.

Comments on the Erdős-Rado Lemma

- A family of more than M sets must contain a sunflower.
- **Plucking** a sunflower means replacing the sets in the sunflower by its core.
- By *repeatedly* finding a sunflower and plucking it, we can reduce a family with more than M sets to a family with at most M sets.
- If $\mathcal{Z}$ is a family of sets, the above result is denoted by pluck($\mathcal{Z}$).
- pluck($\mathcal{Z}$) is not unique.[a]

[a] It depends on the sequence of sunflowers one plucks. Fortunately, this issue is not material to the proof.

An Example of Plucking

- Recall the sunflower on p. 807:

$$\begin{aligned} \mathcal{Z} &= \{\{1,2,3,5\},\{1,2,6,9\},\{0,1,2,11\},\\ &\quad \{1,2,12,13\},\{1,2,8,10\},\{1,2,4,7\}\} \end{aligned}$$

- Then

$$\text{pluck}(\mathcal{Z}) = \{\{1,2\}\}.$$

Razborov's Theorem

Theorem 89 (Razborov, 1985) *There is a constant c such that for large enough n, all monotone circuits for* $\text{CLIQUE}_{n,k}$ *with $k = n^{1/4}$ have size at least $n^{cn^{1/8}}$.*

- We shall approximate any monotone circuit for $\text{CLIQUE}_{n,k}$ by a restricted kind of crude circuit.
- The approximation will proceed in steps: one step for each gate of the monotone circuit.
- Each step introduces few errors (false positives and false negatives).
- Yet, the final crude circuit has exponentially many errors.

The Proof

- Fix $k = n^{1/4}$.
- Fix $\ell = n^{1/8}$.
- Note that[a]
$$2\binom{\ell}{2} \leq k - 1.$$
- p will be fixed later to be $n^{1/8} \log n$.
- Fix $M = (p-1)^\ell \ell!$.
  – Recall the Erdős-Rado lemma (p. 808).

[a] Corrected by Mr. Moustapha Bande (`D98922042`) on January 5, 2010.

The Proof (continued)

- Each crude circuit used in the approximation process is of the form $\text{CC}(X_1, X_2, \ldots, X_m)$, where:
  - $X_i \subseteq V$.
  - $|X_i| \leq \ell$.
  - $m \leq M$.
- It answers true if and only if at least one X_i is a clique.
- We shall show how to approximate any monotone circuit for $\textsc{clique}_{n,k}$ by such a crude circuit, inductively.
- The induction basis is straightforward:
  - Input gate g_{ij} is the crude circuit $\text{CC}(\{i, j\})$.

The Proof (continued)

- A monotone circuit is the OR or AND of two subcircuits.
- We will build approximators of the overall circuit from the approximators of the two subcircuits.
  - Start with two crude circuits $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$.
  - $\mathcal{X}$ and $\mathcal{Y}$ are two families of at most M sets of nodes, each set containing at most ℓ nodes.
  - We will construct the approximate OR and the approximate AND of these subcircuits.
  - Then show both approximations introduce few errors.

The Proof: OR

- $\mathrm{CC}(\mathcal{X} \cup \mathcal{Y})$ is *equivalent to* the OR of $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$.
  - For any node set $\mathcal{C}$, $\mathcal{C} \in \mathcal{X} \cup \mathcal{Y}$ if and only if $\mathcal{C} \in \mathcal{X}$ or $\mathcal{C} \in \mathcal{Y}$.
  - Hence $\mathcal{X} \cup \mathcal{Y}$ contains a clique if and only if $\mathcal{X}$ or $\mathcal{Y}$ contains a clique.
- Problem with $\mathrm{CC}(\mathcal{X} \cup \mathcal{Y})$ occurs when $|\, \mathcal{X} \cup \mathcal{Y} \,| > M$.
- Such violations are eliminated by using

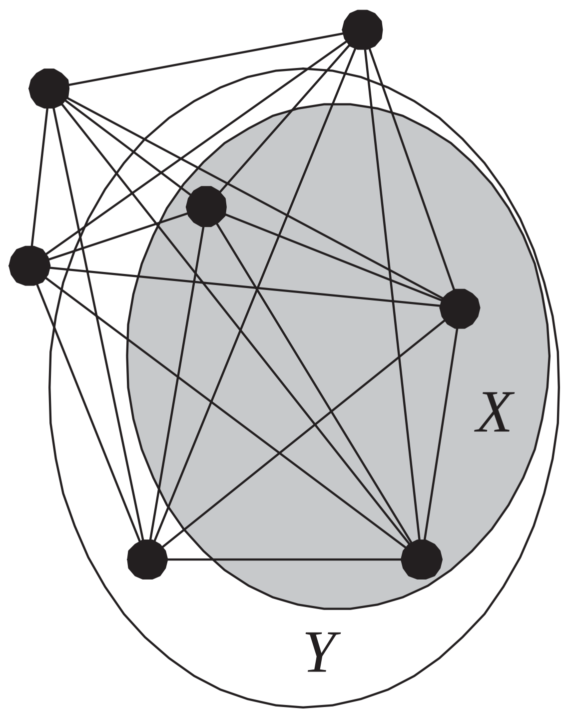
$$\mathrm{CC}(\mathrm{pluck}(\mathcal{X} \cup \mathcal{Y}))$$

  as the final approximate OR of $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$.

The Proof: OR (continued)

- If CC($\mathcal{Z}$) is true, then CC(pluck($\mathcal{Z}$)) must be true.
  - The quick reason: If Y is a clique, then a subset of Y must also be a clique.
  - Let $Y \in \mathcal{Z}$ be a clique.
  - There must exist an $X \in \text{pluck}(\mathcal{Z})$ such that $X \subseteq Y$.
  - This X is also a clique.

The Proof: OR (continued)



The Proof: OR (concluded)

- $\mathrm{CC}(\mathrm{pluck}(\mathcal{X} \cup \mathcal{Y}))$ *introduces* a **false positive** if a negative example makes both $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$ return false but makes $\mathrm{CC}(\mathrm{pluck}(\mathcal{X} \cup \mathcal{Y}))$ return true.
- $\mathrm{CC}(\mathrm{pluck}(\mathcal{X} \cup \mathcal{Y}))$ *introduces* a **false negative** if a positive example makes either $\mathrm{CC}(\mathcal{X})$ or $\mathrm{CC}(\mathcal{Y})$ return true but makes $\mathrm{CC}(\mathrm{pluck}(\mathcal{X} \cup \mathcal{Y}))$ return false.
- We next count the number of false positives and false negatives introduced[a] by $\mathrm{CC}(\mathrm{pluck}(\mathcal{X} \cup \mathcal{Y}))$.
- Let us work on false negatives for OR first.

---

[a]Compared with $\mathrm{CC}(\mathcal{X} \cup \mathcal{Y})$ of course.

The Number of False Negatives[a]

Lemma 90 CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces no false negatives.*

- Each plucking replaces sets in a crude circuit by their common subset.
- This makes the test for cliqueness less stringent.[b]

---

[a]Recall that CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) introduces a false negative if a positive example makes either CC($\mathcal{X}$) or CC($\mathcal{Y}$) return true but makes CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) return false.

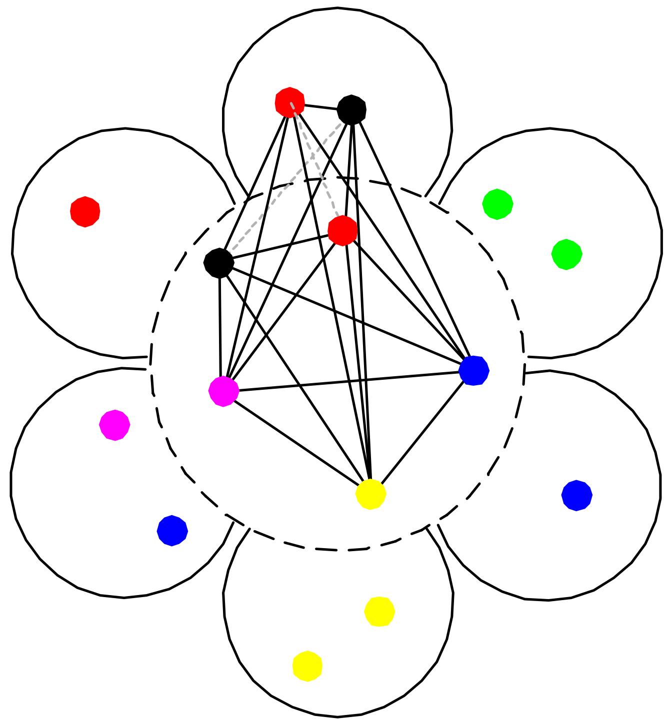
[b]The new crude circuit is at least as positive as the original one (p. 819).

The Number of False Positives

Lemma 91 CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces at most* $\frac{2M}{p-1}\, 2^{-p}(k-1)^n$ *false positives.*

- Each plucking operation replaces the sunflower $\{ Z_1, Z_2, \ldots, Z_p \}$ with its common core Z.
- A false positive is *necessarily* a coloring such that:
  - There is a pair of identically colored nodes in *each* petal Z_i (and so $\text{CC}(Z_1, Z_2, \ldots, Z_p)$ returns false).
  - But the core contains distinctly colored nodes (thus forming a clique).
  - This implies at least one node from each identical-color pair was plucked away.

Proof of Lemma 91 (continued)



Proof of Lemma 91 (continued)

- We now count the number of such colorings.
- Color nodes in V at random with $k-1$ colors.
- Let $R(X)$ denote the event that there are repeated colors in set X.

Proof of Lemma 91 (continued)

- Now

$$
\begin{aligned}
& \text{prob}[\, R(Z_1) \wedge \cdots \wedge R(Z_p) \wedge \neg R(Z)\,] && (24)\\
\leq\ & \text{prob}[\, R(Z_1) \wedge \cdots \wedge R(Z_p) \,|\, \neg R(Z)\,] \\
=\ & \prod_{i=1}^{p} \text{prob}[\, R(Z_i) \,|\, \neg R(Z)\,] \\
\leq\ & \prod_{i=1}^{p} \text{prob}[\, R(Z_i)\,]. && (25)
\end{aligned}
$$

  - Equality holds because $R(Z_i)$ are independent given $\neg R(Z)$ as core Z contains their *only common* nodes.
  - Last inequality holds as the likelihood of repetitions in Z_i decreases given no repetitions in a subset, Z.

Proof of Lemma 91 (continued)

- Consider two nodes in Z_i.
- The probability that they have identical color is
$$\frac{1}{k-1}.$$
- Now
$$\text{prob}[\,R(Z_i)\,] \leq \frac{\binom{|Z_i|}{2}}{k-1} \leq \frac{\binom{\ell}{2}}{k-1} \leq \frac{1}{2}. \tag{26}$$
- So the probability[a] that a random coloring yields a *new* false positive is at most 2^{-p} by inequality (25) on p. 826.

[a]Proportion, if you so prefer.

Proof of Lemma 91 (continued)

- As there are $(k-1)^n$ different colorings, *each* plucking introduces at most $2^{-p}(k-1)^n$ false positives.
- Recall that $|\,\mathcal{X} \cup \mathcal{Y}\,| \leq 2M$.
- When the procedure pluck$(\mathcal{X} \cup \mathcal{Y})$ ends, the set system contains $\leq M$ sets.

Proof of Lemma 91 (concluded)

- Each plucking reduces the number of sets by $p - 1$.
- Hence at most $2M/(p-1)$ pluckings occur in pluck$(\mathcal{X} \cup \mathcal{Y})$.
- At most

$$\frac{2M}{p-1}\, 2^{-p}(k-1)^n$$

false positives are introduced.[a]

---

[a]Note that the numbers of errors are added not multiplied. Recall that we count how many *new* errors are introduced by each approximation step. Contributed by Mr. Ren-Shuo Liu (D98922016) on January 5, 2010.

The Proof: AND

- The approximate AND of crude circuits $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$ is

$$\mathrm{CC}(\mathrm{pluck}(\{\, X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |\, X_i \cup Y_j \,| \leq \ell \,\})).$$

- We need to count the number of errors this approximate AND introduces on the positive and negative examples.

The Proof: AND (continued)

- The approximate AND *introduces* a **false positive** if a negative example makes either CC($\mathcal{X}$) or CC($\mathcal{Y}$) return false but makes the approximate AND return true.
- The approximate AND *introduces* a **false negative** if a positive example makes both CC($\mathcal{X}$) and CC($\mathcal{Y}$) return true but makes the approximate AND return false.
- Introduction of errors means we ignore scenarios where the AND of CC($\mathcal{X}$) and CC($\mathcal{Y}$) is already wrong.

The Proof: AND (continued)

- $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false positives and no false negatives over our positive and negative examples.[a]
  - Suppose $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ returns true.
  - Then some $X_i \cup Y_j$ is a clique.
  - Thus $X_i \in \mathcal{X}$ and $Y_j \in \mathcal{Y}$ are cliques, making both $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$ return true.
  - So $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false positives.

---

[a] Unlike the OR case on p. 818, we are not claiming that $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ is *equivalent to* the AND of $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$. Equivalence is more than we need in either case.

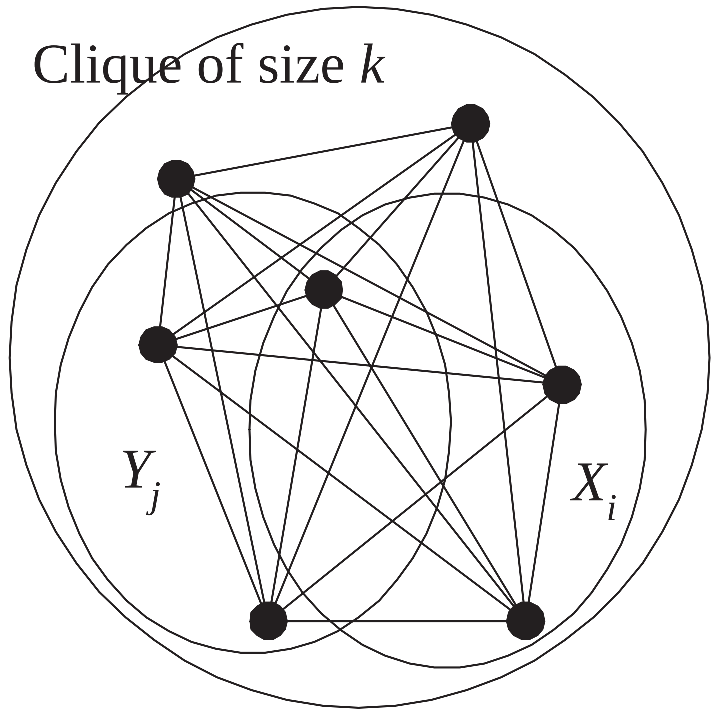
The Proof: AND (concluded)

- (continued)
  - On the other hand, suppose *both* $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$ accept a positive example with a clique $\mathcal{C}$ of size k.
  - This clique $\mathcal{C}$ must contain an $X_i \in \mathcal{X}$ and a $Y_j \in \mathcal{Y}$.
  - As this clique $\mathcal{C}$ also contains $X_i \cup Y_j$,[a] the new circuit returns true.
  - $\mathrm{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false negatives.
- We now bound the number of false positives and false negatives introduced[b] by the approximate AND.

---

[a] See next page.

[b] Compared with $\mathrm{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ of course.


Clique of size k
Y_j
X_i

The Number of False Positives

Lemma 92 *The approximate* AND *introduces at most* $M^2 2^{-p}(k-1)^n$ *false positives.*

- We prove this claim in stages.
- We already knew $\mathrm{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \})$ introduces no false positives.[a]
- $\mathrm{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, | X_i \cup Y_j | \leq \ell \})$ introduces no *additional* false positives because we are testing potentially *fewer* sets for cliqueness.

---

[a]Recall p. 832.

Proof of Lemma 92 (concluded)

- $|\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, | X_i \cup Y_j | \leq \ell \}| \leq M^2$.
- Each plucking reduces the number of sets by $p - 1$.
- So pluck($\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, | X_i \cup Y_j | \leq \ell \}$) involves $\leq M^2/(p-1)$ pluckings.
- Each plucking introduces at most $2^{-p}(k-1)^n$ false positives by the proof of Lemma 91 (p. 823).
- The desired upper bound is

$$[\, M^2/(p-1) \,]\, 2^{-p}(k-1)^n \leq M^2 2^{-p}(k-1)^n.$$

The Number of False Negatives

Lemma 93 *The approximate* AND *introduces at most* $M^2\binom{n-\ell-1}{k-\ell-1}$ *false negatives.*

- We again prove this claim in stages.
- We knew $\text{CC}(\{\, X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y} \,\})$ introduces no false negatives.[a]

---

[a] Recall p. 832.

Proof of Lemma 93 (continued)

- $\text{CC}(\{ X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |\, X_i \cup Y_j \,| \leq \ell \})$ introduces $\leq M^2 \binom{n-\ell-1}{k-\ell-1}$ false negatives.
  - Deletion of set $Z \stackrel{\Delta}{=} X_i \cup Y_j$ larger than ℓ introduces false negatives *only if* Z is part of a clique.
  - There are $\binom{n-|\, Z \,|}{k-|\, Z \,|}$ such cliques.
    * It is the number of positive examples whose clique contains Z.
  - $\binom{n-|\, Z \,|}{k-|\, Z \,|} \leq \binom{n-\ell-1}{k-\ell-1}$ as $|\, Z \,| > \ell$.
  - There are at most M^2 such Zs.

Proof of Lemma 93 (concluded)

- Plucking introduces no false negatives.
  - Recall that if $\text{CC}(\mathcal{Z})$ is true, then $\text{CC}(\text{pluck}(\mathcal{Z}))$ must be true.[a]

[a] Recall p. 819.

Two Summarizing Lemmas

From Lemmas 91 (p. 823) and 92 (p. 835), we have:

Lemma 94 *Each approximation step introduces at most* $M^2 2^{-p}(k-1)^n$ *false positives.*

From Lemmas 90 (p. 822) and 93 (p. 837), we have:

Lemma 95 *Each approximation step introduces at most* $M^2 \binom{n-\ell-1}{k-\ell-1}$ *false negatives.*

The Proof (continued)

- The above two lemmas show that each approximation step introduces "few" false positives and false negatives.
- We next show that the resulting crude circuit has "a lot" of false positives or false negatives.

The Final Crude Circuit

Lemma 96 *Every final crude circuit is:*

1. *Identically false—thus wrong on all positive examples.*
2. *Or outputs true on at least half of the negative examples.*

- Suppose it is not identically false.
- By construction, it accepts at least those graphs that have a clique on some set X of nodes, with

$$|X| \leq \ell = n^{1/8} < n^{1/4} = k.$$

Proof of Lemma 96 (concluded)

- Inequality (26) (p. 827) says that at least half of the colorings assign different colors to nodes in X.
- So at least half of the colorings — thus negative examples — have a clique in X and are accepted.

The Proof (continued)

- Recall the constants on p. 815:

$$\begin{aligned} k &\stackrel{\Delta}{=} n^{1/4}, \\ \ell &\stackrel{\Delta}{=} n^{1/8}, \\ p &\stackrel{\Delta}{=} n^{1/8} \log n, \\ M &\stackrel{\Delta}{=} (p-1)^{\ell} \ell! < n^{(1/3)n^{1/8}} \quad \text{for large } n. \end{aligned}$$

The Proof (continued)

- Suppose the final crude circuit is identically false.
  - By Lemma 95 (p. 840), each approximation step introduces at most $M^2\binom{n-\ell-1}{k-\ell-1}$ false negatives.
  - There are $\binom{n}{k}$ positive examples.
  - The original monotone circuit for $\text{CLIQUE}_{n,k}$ has at least

$$\frac{\binom{n}{k}}{M^2\binom{n-\ell-1}{k-\ell-1}} \geq \frac{1}{M^2}\left(\frac{n-\ell}{k}\right)^{\ell} \geq n^{(1/12)n^{1/8}}$$

gates for large n.

The Proof (concluded)

- Suppose the final crude circuit is not identically false.
  - Lemma 96 (p. 842) says that there are at least $(k-1)^n/2$ false positives.
  - By Lemma 94 (p. 840), each approximation step introduces at most $M^2 2^{-p}(k-1)^n$ false positives
  - The original monotone circuit for $\text{CLIQUE}_{n,k}$ has at least

$$\frac{(k-1)^n/2}{M^2 2^{-p}(k-1)^n} = \frac{2^{p-1}}{M^2} \geq n^{(1/3)n^{1/8}}$$

gates.

Alexander Razborov (1963–)

P $\neq$ NP Proved?

- Razborov's theorem says that there is a monotone language in NP that has no polynomial monotone circuits.
- If we can prove that all monotone languages in P have polynomial monotone circuits, then P $\neq$ NP.
- But Razborov proved in 1985 that some monotone languages in P have no polynomial monotone circuits!