Exponential Circuit Complexity for NP-Complete Problems

- Almost all boolean functions require $\frac{2^n}{2n}$ gates to compute (generalized Theorem 16 on p. 157).
- Progress of using circuit complexity to prove exponential lower bounds for NP-complete problems has been slow.
- We shall prove exponential lower bounds for NP-complete problems using *monotone* circuits.
  - Monotone circuits are circuits without $\neg$ gates.
- Note that this does not settle the P vs. NP problem or any of the conjectures on p. 430.

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. 242).
- 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.
  - The 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 nonmonotone 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 subcircuit 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 *any* of $X_i \subseteq V$ forms a clique.
  - The above-mentioned circuit is $\text{CC}(S_1, S_2, \ldots, S_{\binom{n}{k}})$.

Razborov's Theorem

Theorem 79 (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).
- But the resulting crude circuit has exponentially many errors.

The Proof

- Fix $k = n^{1/4}$.
- Fix $\ell = n^{1/8}$.
- Note that

$$2\binom{\ell}{2} \leq k.$$

- p will be fixed later to be $n^{1/8} \log n$.
- Fix $M = (p-1)^{\ell} \ell!$.
  - Recall the Erdős-Rado lemma (p. 548).

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:
  - $X_i \subseteq V$.
  - $|X_i| \leq \ell$.
  - $m \leq M$.
- We shall show how to approximate any circuit for $\text{CLIQUE}_{n,k}$ 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)

- Any monotone circuit can be considered the OR or AND of two subcircuits.
- We shall show how to build approximators of the overall circuit from the approximators of the two subcircuits.
  - We are given 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 construct the approximate OR and the approximate AND of these subcircuits.
  - Then show both approximations introduce few errors.

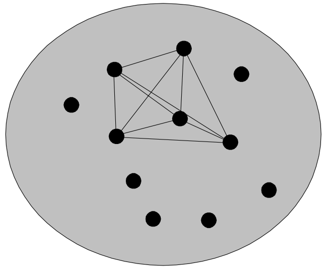
The Proof: Positive Examples

- Error analysis will be applied to only **positive examples** and **negative examples**.
- 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 $\text{CLIQUE}_{n,k}$.

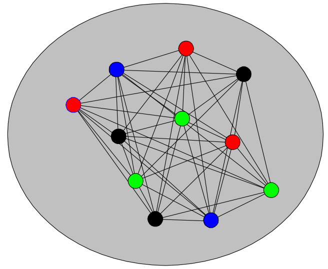
The Proof: Negative Examples

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

Positive and Negative Examples with $k = 5$



A positive example



A negative example

The Proof: OR

- $\text{CC}(\mathcal{X} \cup \mathcal{Y})$ is *equivalent to* the OR of $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$.
- Violations occur when $|\mathcal{X} \cup \mathcal{Y}| > M$.
- Such violations can be eliminated by using

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

  as the approximate OR of $\text{CC}(\mathcal{X})$ and $\text{CC}(\mathcal{Y})$.
- We now count the numbers of errors this approximate OR makes on the positive and negative examples.

The Proof: OR (concluded)

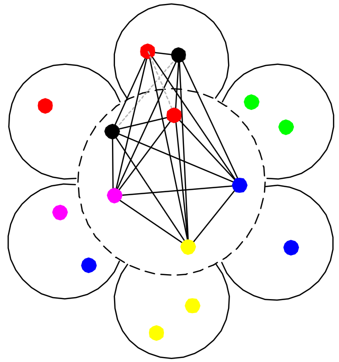
- CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces* a **false positive** if a negative example makes both CC($\mathcal{X}$) and CC($\mathcal{Y}$) return false but makes CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) return true.
- CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces* a **false negative** 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.
- How many false positives and false negatives are introduced by CC(pluck($\mathcal{X} \cup \mathcal{Y}$))?

The Number of False Positives

Lemma 80 CC(pluck($\mathcal{X} \cup \mathcal{Y}$)) *introduces at most* $\frac{M}{p-1} 2^{-p}(k-1)^n$ *false positives.*

- Assume a plucking replaces the sunflower $\{Z_1, Z_2, \ldots, Z_p\}$ with its 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 both crude circuits return false).
  - But the core contains distinctly colored nodes.
    - This implies at least one node from each same-color pair was plucked away.
- We now count the number of such colorings.

Proof of Lemma 80 (continued)



Proof of Lemma 80 (continued)

- Color nodes V at random with $k-1$ colors and let $R(X)$ denote the event that there are repeated colors in set X.
- Now prob$[R(Z_1) \wedge \cdots \wedge R(Z_p) \wedge \neg R(Z)]$ is at most

$$\begin{aligned} & \text{prob}[R(Z_1) \wedge \cdots \wedge R(Z_p) \mid \neg R(Z)] \\ = & \prod_{i=1}^{p} \text{prob}[R(Z_i) \mid \neg R(Z)] \leq \prod_{i=1}^{p} \text{prob}[R(Z_i)]. \quad (7) \end{aligned}$$

  - First equality holds because $R(Z_i)$ are independent given $\neg R(Z)$ as Z contains their only common nodes.
  - Last inequality holds as the likelihood of repetitions in Z_i decreases given no repetitions in $Z \subseteq Z_i$.

Proof of Lemma 80 (continued)

- Consider two nodes in Z_i.
- The probability that they have identical color is $\frac{1}{k-1}$.
- Now $\text{prob}[\,R(Z_i)\,] \leq \frac{\binom{|Z_i|}{2}}{k-1} \leq \frac{\binom{\ell}{2}}{k-1} \leq \frac{1}{2}$.
- So the probability that a random coloring is a new false positive is at most 2^{-p} by inequality (7).
- As there are $(k-1)^n$ different colorings, each plucking introduces at most $2^{-p}(k-1)^n$ false positives.

Proof of Lemma 80 (concluded)

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

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

false positives are introduced.

The Number of False Negatives

Lemma 81 $\text{CC}(\text{pluck}(\mathcal{X} \cup \mathcal{Y}))$ *introduces no false negatives.*

- Each plucking replaces a set in a crude circuit by a subset.
- This makes the test less stringent.
  - For each $Y \in \mathcal{X} \cup \mathcal{Y}$, there must exist at least one $X \in \text{pluck}(\mathcal{X} \cup \mathcal{Y})$ such that $X \subseteq Y$.
  - So if $Y \in \mathcal{X} \cup \mathcal{Y}$ is a clique, then $\text{pluck}(\mathcal{X} \cup \mathcal{Y})$ also contains a clique in X.
- So plucking can only increase the number of accepted graphs.

The Proof: AND

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

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

- We now count the numbers of errors this approximate AND makes on the positive and negative examples.

The Proof: AND (concluded)

- 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.
- How many false positives and false negatives are introduced by the approximate AND?

The Number of False Positives

Lemma 82 *The approximate* AND *introduces at most* $M^2 2^{-p}(k-1)^n$ *false positives.*

- $CC(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}\})$ introduces no false positives.
  - If $X_i \cup Y_j$ is a clique, both X_i and Y_j must be cliques, making both $CC(\mathcal{X})$ and $CC(\mathcal{Y})$ return true.
- $CC(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell\})$ introduces no false positives for the same reason as above.

Proof of Lemma 82 (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 80 (p. 566).
- 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 83 *The approximate* AND *introduces at most* $M^2\binom{n-\ell-1}{k-\ell-1}$ *false negatives.*

- We follow the same three-step proof as before.
- $CC(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}\})$ introduces no false negatives.
  - Suppose both $CC(\mathcal{X})$ and $CC(\mathcal{Y})$ accept a positive example with a clique of size k.
  - The clique must contain an $X_i \in \mathcal{X}$ and a $Y_j \in \mathcal{Y}$.
  - As it contains $X_i \cup Y_j$, the new circuit returns true.

Proof of Lemma 83 (concluded)

- $\mathrm{CC}(\{X_i \cup Y_j : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, |X_i \cup Y_j| \leq \ell\})$ introduces $\leq M^2 \binom{n-\ell-1}{k-\ell-1}$ false negatives.
  - Deletion of set $Z = X_i \cup Y_j$ larger than ℓ introduces false negatives which are cliques containing Z.
  - There are $\binom{n-|Z|}{k-|Z|}$ such cliques.
  - $\binom{n-|Z|}{k-|Z|} \leq \binom{n-\ell-1}{k-\ell-1}$ as $|Z| \geq \ell$.
  - There are at most M^2 such Zs.
- Plucking introduces no false negatives.

Two Summarizing Lemmas

From Lemmas 80 (p. 566) and 82 (p. 574), we have:

Lemma 84 *Each approximation step introduces at most* $M^2 2^{-p} (k-1)^n$ *false positives.*

From Lemmas 81 (p. 571) and 83 (p. 576), we have:

Lemma 85 *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 introduce "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 86 *Every final crude circuit either is identically false—thus wrong on all positive examples—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 $|X| \leq \ell$, which at $n^{1/8}$ is less than $k = n^{1/4}$.
- The proof of Lemma 80 (p. 566ff) shows that at least half of the colorings assign different colors to nodes in X.
- So half of the negative examples have a clique in X and are accepted.

The Proof (continued)

- Recall the constants on p. 558: $k = n^{1/4}$, $\ell = n^{1/8}$, $p = n^{1/8} \log n$, $M = (p-1)^\ell \ell! < n^{(1/3) n^{1/8}}$ for large n.
- Suppose the final crude circuit is identically false.
  - By Lemma 85 (p. 578), 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 crude 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 86 (p. 580) says that there are at least $(k-1)^n/2$ false positives.
  - By Lemma 84 (p. 578), each approximation step introduces at most $M^2 2^{-p} (k-1)^n$ false positives
  - The original crude 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.

P ≠ 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 ≠ NP.
- But Razborov proved in 1985 that some monotone languages in P have no polynomial monotone circuits!

PSPACE and Games

- Given a boolean expression ϕ in CNF with boolean variables $x_1, x_2, \ldots, x_n$, is it true that $\exists x_1 \forall x_2 \cdots Q_n x_n \phi$?
- This is called **quantified satisfiability** or QSAT.
- This problem is like a two-person game: $\exists$ and $\forall$ are the two players.
- We ask then is there a winning strategy for $\exists$?

QSAT $\in$ PSPACE

QSAT$(Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \phi(x_1, \ldots, x_n))$:
if $n = 0$ **then**
  return ϕ;
else
  if $Q_1 = \exists$ **then**
    return QSAT$(Q_2 x_2 \cdots Q_n x_n \phi(0, x_2, \ldots, x_2)) \vee$
    QSAT$(Q_2 x_2 \cdots Q_n x_n \phi(1, x_2, \ldots, x_2))$;
  else
    return QSAT$(Q_2 x_2 \cdots Q_n x_n \phi(0, x_2, \ldots, x_2)) \wedge$
    QSAT$(Q_2 x_2 \cdots Q_n x_n \phi(1, x_2, \ldots, x_2))$;
  end if
end if

IP and PSPACE

- We next prove that coNP $\subseteq$ IP.
- Shamir in 1990 proved that IP equals PSPACE using similar ideas.

Theorem 87 *IP = PSPACE.*

Interactive Proof for Boolean Unsatisfiability

- A 3SAT formula is a conjunction of disjunctions of at most three literals.
- We shall present an interactive proof for boolean unsatisfiability.
- For any unsatisfiable 3SAT formula $\phi(x_1, x_2, \ldots, x_n)$, there is an interactive proof for the fact that it is unsatisfiable.
- Therefore, coNP $\subseteq$ IP.

Arithmetization of Boolean Formulas

The idea is to arithmetize the boolean formula.

- T $\to$ positive integer
- F $\to$ 0
- $x_i \to x_i$
- $\bar{x_i} \to 1 - x_i$
- $\vee \to +$
- $\wedge \to \times$
- $\phi(x_1, x_2, \ldots, x_n) \to \Phi(x_1, x_2, \ldots, x_n)$

The Arithmetic Version

- A boolean formula is transformed into a multivariate polynomial Φ.
- It is easy to verify that ϕ is unsatisfiable if and only if

$$\sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, x_2, \ldots, x_n) = 0.$$

- But the above seems to require exponential time.
- We turn to more intricate methods.

Choosing the Field

- Suppose ϕ has m clauses of length three each.
- Then $\Phi(x_1, x_2, \ldots, x_n) \leq 3^m$ for any truth assignment $(x_1, x_2, \ldots, x_n)$.
- Because there are at most 2^n truth assignments,

$$\sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, x_2, \ldots, x_n) \leq 2^n 3^m.$$

Choosing the Field (concluded)

- By choosing a prime $q > 2^n 3^m$ and working modulo this prime, proving unsatisfiability reduces to proving that

$$\sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_n=0,1} \Phi(x_1, x_2, \ldots, x_n) \equiv 0 \bmod q.$$

- Working under a *finite* field allows us to uniformly select a random element in the field.

Binding Peggy

- Peggy has to find a sequence of polynomials that satisfy a number of restrictions.
- The restrictions are imposed by Victor: After receiving a polynomial from Peggy, Victor sets a new restriction for the next polynomial in the sequence.
- These restrictions guarantee that if ϕ is unsatisfiable, such a sequence can always be found.
- However, if ϕ is not unsatisfiable, any Peggy has only a small probability of finding such a sequence.
  - The probability is taken over Victor's coin tosses.

The Algorithm

```
Peggy and Victor both arithmetize $\phi$ to obtain $\Phi$;
Peggy picks a prime $q > 2^n 3^m$ and sends it to Victor;
Victor rejects and stops if $q$ is not a prime;
Victor sets $v_0$ to 0;
for $i = 1, 2, \ldots, n$ do
    Peggy calculates $P_i^*(z) =$
    $\sum_{x_{i+1}=0,1} \cdots \sum_{x_n=0,1} \Phi(r_1, \ldots, r_{i-1}, z, x_{i+1}, \ldots, x_n)$;
    Peggy sends $P_i^*(z)$ to Victor;
    Victor rejects and stops if $P_i^*(0) + P_i^*(1) \not\equiv v_{i-1} \bmod q$ or
    $P_i^*(z)$'s degree exceeds $m$; {$P_i^*(z)$ has at most $m$ clauses.}
    Victor uniformly picks $r_i \in Z_q$ and calculates $v_i = P_i^*(r_i)$;
    Victor sends $r_i$ to Peggy;
end for
Victor accepts iff $\Phi(r_1, r_2, \ldots, r_n) \equiv v_n \bmod q$;
```