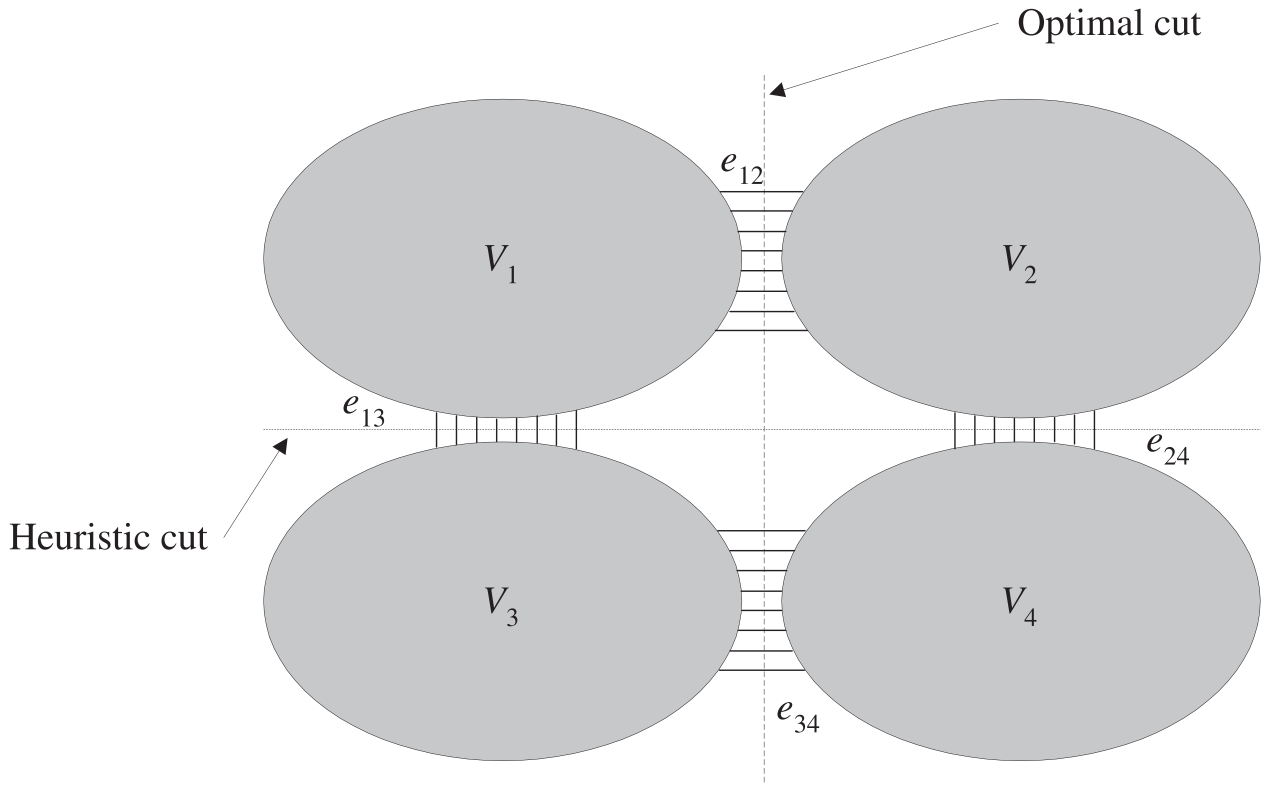
MAX CUT Revisited

- The NP-complete MAX CUT seeks to partition the nodes of graph $G = (V, E)$ into $(S, V - S)$ so that there are as many edges as possible between S and $V - S$ (p. 216).
- **Local search** is a heuristic that starts from any feasible solution and performs a "local" improvement until no improvements are possible.

A 0.5-Approximation Algorithm for MAX CUT

```
S := ∅;
while ∃v ∈ V whose switching sides results in a larger cut do
    S := S ∪ {v};
end while
return S;
```

The Analysis



Analysis (continued)

- Partition $V = V_1 \cup V_2 \cup V_3 \cup V_4$, where our algorithm returns $(V_1 \cup V_2, V_3 \cup V_4)$ and the optimum cut is $(V_1 \cup V_3, V_2 \cup V_4)$.
- Let e_{ij} be the number of edges between V_i and V_j.
- Because no migration of nodes can improve the algorithm's cut, for each node in V_1, its edges to $V_1 \cup V_2$ are outnumbered by those to $V_3 \cup V_4$.
- Considering all nodes in V_1 together, we have $2e_{11} + e_{12} \leq e_{13} + e_{14}$, which implies

$$e_{12} \leq e_{13} + e_{14}.$$

Analysis (continued)

- Similarly,

$$\begin{aligned} e_{12} &\leq e_{23} + e_{24} \\ e_{34} &\leq e_{23} + e_{13} \\ e_{34} &\leq e_{14} + e_{24} \end{aligned}$$

- Adding all four inequalities, dividing both sides by 2, and adding the inequality $e_{14} + e_{23} \leq e_{14} + e_{23} + e_{13} + e_{24}$, we obtain

$$e_{12} + e_{34} + e_{14} + e_{23} \leq 2(e_{13} + e_{14} + e_{23} + e_{24}).$$

- The above says our solution is at least half the optimum.

Unapproximability of TSP

- Algorithms with an approximation threshold less than 1 have been exhibited for NODE COVER, MAXSAT, and MAX CUT.
- The situation is maximally pessimistic for TSP: It cannot be approximated unless P = NP.

Theorem 68 *The approximation threshold of* TSP *is 1 unless $P = NP$, when it becomes 0.*

The Proof

- Suppose that there is a polynomial-time ϵ-approximation algorithm for TSP for some $\epsilon < 1$.
- We shall construct a polynomial-time algorithm for the NP-complete HAMILTONIAN CYCLE.
- Given any graph $G = (V, E)$, construct a TSP with $|V|$ cities with distances

$$d_{ij} = \begin{cases} 1, & \text{if } [\,i, j\,] \in E \\ \frac{|V|}{1-\epsilon}, & \text{otherwise} \end{cases}$$

- Run the alleged approximation algorithm on this TSP instance.

The Proof (continued)

- Suppose that a tour of cost $|V|$ is returned.
  - This tour must be a Hamiltonian cycle.
- Suppose that a tour with at least one edge of length $\frac{|V|}{1-\epsilon}$ is returned.
  - The total length of this tour is $> \frac{|V|}{1-\epsilon}$.
  - Because the algorithm is ϵ-approximate, the optimum is at least $1 - \epsilon$ times the returned tour's length.
  - The optimum tour has a cost exceeding $|V|$.
  - Hence G has no Hamiltonian cycles.

KNAPSACK Has an Approximation Threshold of Zero

Theorem 69 *For any ϵ, there is a polynomial-time ϵ-approximation algorithm for* KNAPSACK.

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

$$V = \max\{v_1, v_2, \ldots, v_n\}.$$

The Proof (continued)

- For $0 \leq i \leq n$ and $0 \leq v \leq nV$, define $W(i, v)$ to be the minimum weight attainable by selecting some among the i first items, so that their value is exactly v.
- Start with $W(0, v) = \infty$ for all v.
- Then $W(i+1, v) = \min\{W(i, v), W(i, v - v_{i+1}) + w_{i+1}\}$.
- Finally, pick the largest v such that $W(n, v) \leq W$.
- The running time is $O(n^2 V)$, not exactly polynomial time.
- Next idea: Limit the number of precision bits.

The Proof (continued)

- Given the instance $x = (w_1, \ldots, w_n, W, v_1, \ldots, v_n)$, we define the approximate instance

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

where

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

- Solving x' takes time $O(n^2 V/2^b)$.
- The solution S' is close to the optimum solution S:

$$\sum_{i \in S} v_i \geq \sum_{i \in S'} v_i \geq \sum_{i \in S'} v_i' \geq \sum_{i \in S} v_i' \geq \sum_{i \in S} (v_i - 2^b) \geq \sum_{i \in S} v_i - n2^b.$$

The Proof (continued)

- Hence
$$\sum_{i \in S'} v_i \geq \sum_{i \in S} v_i - n2^b.$$
- Because V is a lower bound on the value of the optimum solution (without loss of generality, $w_i \leq W$), the relative deviation from the optimum is at most $\epsilon = n2^b/V$.
- By truncating the last $b = \lceil \log \frac{\epsilon V}{n} \rceil$ bits of the values, the algorithm becomes ϵ-approximate with running time $O(n^2 V/b) = O(n^3/\epsilon)$, a polynomial.

A Loose End

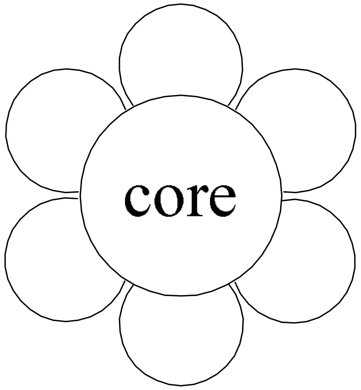
- If V is small, say n, then $\epsilon = 2^b$ and cannot be less than one however $b \in \mathbb{N}$ is picked.
- The remedy is to use the truncation idea only when, say, $V > n^2$.
  - The dynamic-programming algorithm runs in time $O(n^2 V) = O(n^4)$ when $V \leq n^2$.
- Now,

$$b = \lceil \log \frac{\epsilon V}{n} \rceil > \lceil \log n\epsilon \rceil \geq 0$$

for suitably large n.

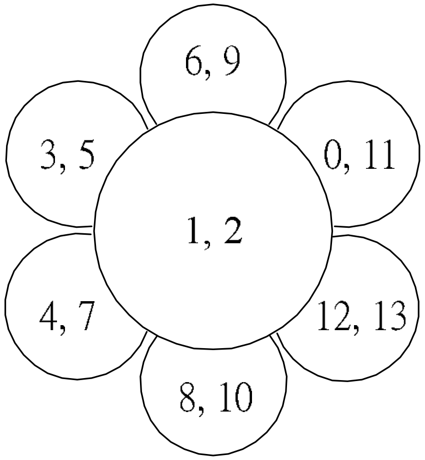
Sunflowers

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

- 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)

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

The Proof of the Erdős-Rado Lemma (continued)

- (continued)
  - $\mathcal{Z}'$ 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 entails 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($\mathcal{Z}$).

Exponential Circuit Complexity for NP-Complete Problems

- Almost all boolean functions require $\frac{2^n}{2n}$ gates to compute (generalized Theorem 9 on p. 110).
- 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. 350.

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. 176).
- There are NP-complete problems that are not monotone; hence they cannot be computed by monotone circuits whatever the sizes.
- There are NP-complete problems that are monotone; hence they can be computed by monotone circuits.
  - HAMILTONIAN PATH and CLIQUE.

CLIQUE$_{n,k}$

- 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.
- CLIQUE$_{n,k}$ is a monotone function.
- Thus it can be computed by a monotone circuit.

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** $\mathrm{CC}(X_1, X_2, \ldots, X_m)$ tests if any of $X_i \subseteq V$ forms a clique.
  - The above-mentioned circuit is $\mathrm{CC}(S_1, S_2, \ldots, S_{\binom{n}{k}})$.

Razborov's Theorem

Theorem 71 (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.

Proof of Razborov's Theorem

- Fix $k = n^{1/4}$.
- Fix $\ell = n^{1/8}$.
- p will be fixed later to be $n^{1/8} \log n$.
- Fix $M = (p-1)^{\ell} \ell!$ (recall the Erdős-Rado Lemma on p. 440).
- Note that

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

Proof of Razborov's Theorem (continued)

- Each crude circuit used in the approximation process is of the form $\mathrm{CC}(X_1, X_2, \ldots, X_m)$, 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 $\mathrm{CC}(\{i, j\})$.

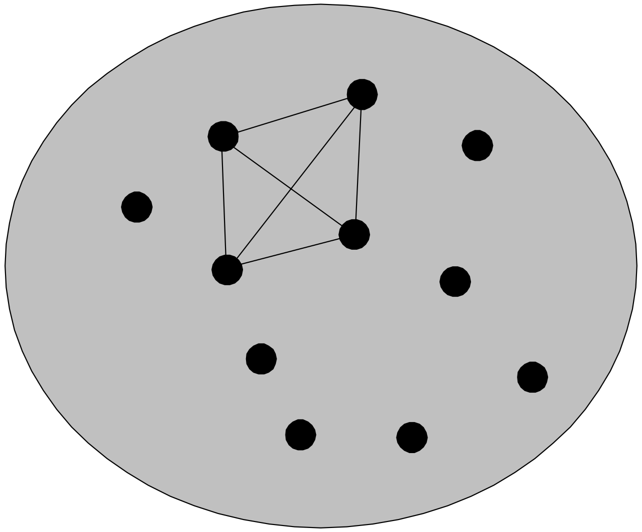
Proof of Razborov's Theorem (continued)

- 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 $\mathrm{CC}(\mathcal{X})$ and $\mathrm{CC}(\mathcal{Y})$.
  - $\mathcal{X}$ 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* circuits.
  - Then show both approximations introduce few errors.

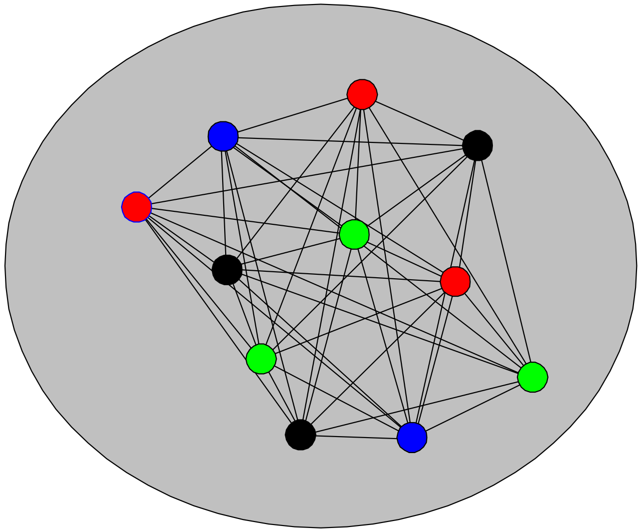
Proof of Razborov's Theorem (continued)

- 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 and they all should elicit a true output from $\text{CLIQUE}_{n,k}$.
- A negative example: 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 and they all should elicit a false output from $\text{CLIQUE}_{n,k}$.

Proof of Razborov's Theorem (continued)



A positive example



A negative example