Generalized 2SAT: MAX2SAT

- Consider a 2SAT formula.
- Let $K \in \mathbb{N}$.
- MAX2SAT asks whether there is a truth assignment that satisfies at least $K$ of the clauses.
  - MAX2SAT becomes 2SAT when $K$ equals the number of clauses.
Generalized 2SAT: MAX2SAT (concluded)

- **MAX2SAT** is an optimization problem.
  - With binary search, one can nail the maximum number of satisfiable clauses of 2SAT formulas.

- **MAX2SAT ∈ NP**: Guess a truth assignment and verify the count.

- We now reduce 3SAT to MAX2SAT.
MAX2SAT Is NP-Complete\textsuperscript{a}

- Consider the following 10 clauses:

\[
(x) \land (y) \land (z) \land (w) \\
(\neg x \lor \neg y) \land (\neg y \lor \neg z) \land (\neg z \lor \neg x) \\
(x \lor \neg w) \land (y \lor \neg w) \land (z \lor \neg w)
\]

- Let the 2SAT formula \( r(x, y, z, w) \) represent the conjunction of these clauses.

- The clauses are symmetric with respect to \( x, y, \) and \( z \).

- How many clauses can we satisfy?

\textsuperscript{a}Garey, Johnson, & Stockmeyer (1976).
The Proof (continued)

All of $x, y, z$ are true: By setting $w$ to true, we satisfy $4 + 0 + 3 = 7$ clauses, whereas by setting $w$ to false, we satisfy only $3 + 0 + 3 = 6$ clauses.

Two of $x, y, z$ are true: By setting $w$ to true, we satisfy $3 + 2 + 2 = 7$ clauses, whereas by setting $w$ to false, we satisfy $2 + 2 + 3 = 7$ clauses.
The Proof (continued)

One of $x, y, z$ is true: By setting $w$ to false, we satisfy $1 + 3 + 3 = 7$ clauses, whereas by setting $w$ to true, we satisfy only $2 + 3 + 1 = 6$ clauses.

None of $x, y, z$ is true: By setting $w$ to false, we satisfy $0 + 3 + 3 = 6$ clauses, whereas by setting $w$ to true, we satisfy only $1 + 3 + 0 = 4$ clauses.
The Proof (continued)

- A truth assignment that satisfies $x \lor y \lor z$ can be extended to satisfy 7 of the 10 clauses of $r(x, y, z, w)$, and no more.

- A truth assignment that does not satisfy $x \lor y \lor z$ can be extended to satisfy only 6 of them, and no more.

- The reduction from 3SAT $\phi$ to MAX2SAT $R(\phi)$:
  - For each clause $C_i = (\alpha \lor \beta \lor \gamma)$ of $\phi$, add group $r(\alpha, \beta, \gamma, w_i)$ to $R(\phi)$.

- If $\phi$ has $m$ clauses, then $R(\phi)$ has $10m$ clauses.
The Proof (continued)

- Finally, set $K = 7m$.

- We now show that $K$ clauses of $R(\phi)$ can be satisfied if and only if $\phi$ is satisfiable.
The Proof (continued)

• Suppose $K = 7m$ clauses of $R(\phi)$ can be satisfied.
  
  – 7 clauses of each group $r(\alpha, \beta, \gamma, w_i)$ must be satisfied because each group can have at most 7 clauses satisfied.\(^a\)

  – Hence each clause $C_i = (\alpha \lor \beta \lor \gamma)$ of $\phi$ is satisfied by the same truth assignment.

  – So $\phi$ is satisfied.

\(^a\)If 70% of the world population are male and if at most 70% of each country’s population are male, then each country must have exactly 70% male population.
The Proof (concluded)

- Suppose \( \phi \) is satisfiable.
  - Let \( T \) satisfy all clauses of \( \phi \).
  - Each group \( r(\alpha, \beta, \gamma, w_i) \) can set its \( w_i \) appropriately to have 7 clauses satisfied.
  - So \( K = 7m \) clauses are satisfied.
NAESAT

• The NAESAT (for “not-all-equal” SAT) is like 3SAT.

• But there must be a satisfying truth assignment under which no clauses have all three literals equal in truth value.

• Equivalently, there is a truth assignment such that each clause has a literal assigned true and a literal assigned false.

• Equivalently, there is a satisfying truth assignment under which each clause has a literal assigned false.
NAESAT (concluded)

• Take

\[ \phi = (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor \neg x_3) \]
\[ \land (x_1 \lor x_2 \lor x_3) \]

as an example.

• Then \( \{ x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{false} \} \) NAE-satisfies \( \phi \) because

\[(\text{false} \lor \text{true} \lor \text{true}) \land (\text{false} \lor \text{false} \lor \text{true}) \]
\[\land (\text{true} \lor \text{false} \lor \text{false}).\]
NAESAT Is NP-Complete\textsuperscript{a}

- Recall the reduction of CIRCUIT SAT to SAT on p. 279ff.
- It produced a CNF $\phi$ in which each clause has 1, 2, or 3 literals.
- Add the same variable $z$ to all clauses with fewer than 3 literals to make it a 3SAT formula.
- Goal: The new formula $\phi(z)$ is NAE-satisfiable if and only if the original circuit is satisfiable.

\textsuperscript{a}Schaefer (1978).
The Proof (continued)

- The following simple observation will be useful.
- Suppose $T$ NAE-satisfies a boolean formula $\phi$.
- Let $\overline{T}$ take the opposite truth value of $T$ on every variable.
- Then $\overline{T}$ also NAE-satisfies $\phi$.\(^\text{a}\)

\(^{\text{a}}\)Hesse’s *Siddhartha* (1922), “The opposite of every truth is just as true!”
The Proof (continued)

• Suppose \( T \) NAE-satisfies \( \phi(z) \).
  
  – \( \overline{T} \) also NAE-satisfies \( \phi(z) \).
  
  – Under \( T \) or \( \overline{T} \), variable \( z \) takes the value false.
  
  – \textit{This} truth assignment \( T \) must satisfy all the clauses of \( \phi \).
    
    * Because \( z \) is not the reason that makes \( \phi(z) \) true under \( T \) anyway.
  
  – So \( T \models \phi \).
  
  – And the original circuit is satisfiable.
The Proof (concluded)

• Suppose there is a truth assignment that satisfies the circuit.
  – Then there is a truth assignment $T$ that satisfies every clause of $\phi$.
  – Extend $T$ by adding $T(z) = \text{false}$ to obtain $T'$.
  – $T'$ satisfies $\phi(z)$.
  – So in no clauses are all three literals false under $T'$.
  – In no clauses are all three literals true under $T'$.
    * Need to go over the detailed construction on pp. 280–282.
Undirected Graphs

- An undirected graph $G = (V, E)$ has a finite set of nodes, $V$, and a set of undirected edges, $E$.
- It is like a directed graph except that the edges have no directions and there are no self-loops.
- Use $[i, j]$ to mean there is an undirected edge between node $i$ and node $j$. 
Independent Sets

- Let $G = (V, E)$ be an undirected graph.
- $I \subseteq V$.
- $I$ is independent if there is no edge between any two nodes $i, j \in I$.
- INDEPENDENT SET: Given an undirected graph and a goal $K$, is there an independent set of size $K$?
- Many applications.
INDEPENDENT SET Is NP-Complete

- This problem is in NP: Guess a set of nodes and verify that it is independent and meets the count.

- We will reduce 3SAT to INDEPENDENT SET.

- If a graph contains a triangle, any independent set can contain at most one node of the triangle.

- The results of the reduction will be graphs whose nodes can be partitioned into disjoint triangles, one for each clause.\(^a\)

\(^a\)Recall that a reduction does not have to be an onto function.
The Proof (continued)

- Let $\phi$ be a 3SAT formula with $m$ clauses.
- We will construct graph $G$ with $K = m$.
- Furthermore, $\phi$ is satisfiable if and only if $G$ has an independent set of size $K$.
- Here is the reduction:
  - There is a triangle for each clause with the literals as the nodes.
  - Add edges between $x$ and $\neg x$ for every variable $x$. 
\[(x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3)\]

Same literal labels that appear in the same clause or different clauses yield *distinct* nodes.
The Proof (continued)

• Suppose $G$ has an independent set $I$ of size $K = m$.
  – An independent set can contain at most $m$ nodes, one from each triangle.
  – So $I$ contains exactly one node from each triangle.
  – Truth assignment $T$ assigns true to those literals in $I$.
  – $T$ is consistent because contradictory literals are connected by an edge; hence both cannot be in $I$.
  – $T$ satisfies $\phi$ because it has a node from every triangle, thus satisfying every clause.\(^a\)

\(^a\)The variables without a truth value can be assigned arbitrarily. Contributed by Mr. Chun-Yuan Chen (R99922119) on November 2, 2010.
The Proof (concluded)

• Suppose $\phi$ is satisfiable.
  
  – Let truth assignment $T$ satisfy $\phi$.
  
  – Collect one node from each triangle whose literal is true under $T$.
  
  – The choice is arbitrary if there is more than one true literal.
  
  – This set of $m$ nodes must be independent by construction.
    
    * Because both literals $x$ and $\neg x$ cannot be assigned true.
Other INDEPENDENT SET-Related NP-Complete Problems

Corollary 42  INDEPENDENT SET is NP-complete for 4-degree graphs.

Theorem 43  INDEPENDENT SET is NP-complete for planar graphs.

Theorem 44 (Garey & Johnson, 1977))  INDEPENDENT SET is NP-complete for 3-degree planar graphs.
Is INDEPENDENT EDGE SET Also NP-Complete?

- INDEPENDENT EDGE SET: Given an undirected graph and a goal $K$, is there an independent edge set of size $K$?
- This problem is equivalent to maximum matching!
- Maximum matching can be solved in polynomial time.\(^a\)

\(^a\)Edmonds (1965); Micali & V. Vazirani (1980).
A Maximum Matching
NODE COVER

- We are given an undirected graph $G$ and a goal $K$.
- NODE COVER: Is there a set $C$ with $K$ or fewer nodes such that each edge of $G$ has at least one of its endpoints (i.e., incident nodes) in $C$?
- Many applications.
NODE COVER (concluded)
Corollary 45 (Karp, 1972) NODE COVER is NP-complete.

- \( I \) is an independent set of \( G = (V, E) \) if and only if \( V - I \) is a node cover of \( G \).
Richard Karp\textsuperscript{a} (1935–)

\textsuperscript{a}Turing Award (1985).
Remarks$^a$

- Are INDEPENDENT SET and NODE COVER in P if $K$ is a constant?
  - Yes, because one can do an exhaustive search on all the possible node covers or independent sets (both $\binom{n}{K}$ of them, a polynomial).$^b$

- Are INDEPENDENT SET and NODE COVER NP-complete if $K$ is a linear function of $n$?
  - INDEPENDENT SET with $K = n/3$ and NODE COVER with $K = 2n/3$ remain NP-complete by our reductions.

$^a$Contributed by Mr. Ching-Hua Yu (D00921025) on October 30, 2012.

$^b$n = $|V|$. 
CLIQUE

- We are given an undirected graph $G$ and a goal $K$.
- CLIQUE asks if there is a set $C$ with $K$ nodes such that there is an edge between any two nodes $i, j \in C$.
- Many applications.
CLIQUE (concluded)
**Corollary 46**  
**CLIQUE is NP-Complete**

CLIQUE is NP-complete.

- Let $\bar{G}$ be the complement of $G$, where $[x, y] \in \bar{G}$ if and only if $[x, y] \notin G$.

- $I$ is a clique in $G$ $\iff$ $I$ is an independent set in $\bar{G}$.

\[\text{\footnotesize{\textsuperscript{a}Karp (1972).}}\]
MIN CUT and MAX CUT

• A cut in an undirected graph $G = (V, E)$ is a partition of the nodes into two nonempty sets $S$ and $V - S$.

• The size of a cut $(S, V - S)$ is the number of edges between $S$ and $V - S$.

• MIN CUT asks for the minimum cut size.

• MIN CUT $\in$ P by the maxflow algorithm.a

• MAX CUT asks if there is a cut of size at least $K$.
  – $K$ is part of the input.

---

aFord & Fulkerson (1962); Orlin (2012) improves the running time to $O(|V| \cdot |E|)$. 
A Cut of Size 4
MIN CUT and MAX CUT (concluded)

- MAX CUT has applications in circuit layout.
  - The minimum area of a VLSI layout of a graph is not less than the square of its maximum cut size.\(^a\)

\(^a\)Raspaud, Sýkora, & Vrťo (1995); Mak & Wong (2000).
MAX CUT Is NP-Complete\(^a\)

- We will reduce NAESAT to MAX CUT.
- Given a 3SAT formula \(\phi\) with \(m\) clauses, we shall construct a graph \(G = (V, E)\) and a goal \(K\).
- Furthermore, there is a cut of size at least \(K\) if and only if \(\phi\) is NAE-satisfiable.
- Our graph will have multiple edges between two nodes.
  - Each such edge contributes one to the cut if its nodes are separated.

\(^a\)Karp (1972); Garey, Johnson, & Stockmeyer (1976). MAX CUT remains NP-complete even for graphs with maximum degree 3 (Makedon, Papadimitriou, & Sudborough, 1985).
The Proof

• Suppose $\phi$’s $m$ clauses are $C_1, C_2, \ldots, C_m$.

• The boolean variables are $x_1, x_2, \ldots, x_n$.

• $G$ has $2n$ nodes: $x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n$.

• Each clause with 3 distinct literals makes a triangle in $G$.

• For each clause with two identical literals, there are two parallel edges between the two distinct literals.
The Proof (continued)

- No need to consider clauses with one literal (why?).
- No need to consider clauses containing two opposite literals $x_i$ and $\neg x_i$ (why?).
- For each variable $x_i$, add $n_i$ copies of edge $[x_i, \neg x_i]$, where $n_i$ is the number of occurrences of $x_i$ and $\neg x_i$ in $\phi$.
- Note that
  \[
  \sum_{i=1}^{n} n_i = 3m.
  \]
  - The summation is simply the total number of literals.
\[ x_i \quad \sim x_j \quad \sim x_k \]
\[ x_i \quad \sim \sim x_j \]
\[ x_i \quad \sim \sim \sim \sim x_i \quad n_i \text{ copies} \]
The Proof (continued)

• Set $K = 5m$.

• Suppose there is a cut $(S, V - S)$ of size $5m$ or more.

• A clause (a triangle or two parallel edges) contributes at most 2 to a cut no matter how you split it.

• Suppose some $x_i$ and $\neg x_i$ are on the same side of the cut.

• They together contribute at most $2n_i$ edges to the cut.
  – They appear in at most $n_i$ different clauses.
  – A clause contributes at most 2 to a cut.
The Proof (continued)

- Either $x_i$ or $\neg x_i$ contributes at most $n_i$ to the cut by the pigeonhole principle.

- Changing the side of that literal does *not decrease* the size of the cut.

- Hence we assume variables are separated from their negations.

- The total number of edges in the cut that join opposite literals $x_i$ and $\neg x_i$ is $\sum_{i=1}^{n} n_i$.

- But we knew $\sum_{i=1}^{n} n_i = 3m$. 
The Proof (concluded)

- The remaining $K - 3m \geq 2m$ edges in the cut must come from the $m$ triangles or parallel edges that correspond to the clauses.

- Each can contribute at most 2 to the cut.

- So all are split.

- A split clause means at least one of its literals is true and at least one false.

- The other direction is left as an exercise.
This Cut Does Not Meet the Goal $K = 5 \times 3 = 15$

- $(x_1 \lor x_2 \lor x_2) \land (x_1 \lor \neg x_3 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)$.
- The cut size is $13 < 15$. 
This Cut Meets the Goal $K = 5 \times 3 = 15$

- $(x_1 \lor x_2 \lor x_2) \land (x_1 \lor \neg x_3 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)$.
- The cut size is now 15.
Remarks

• We had proved that MAX CUT is NP-complete for multigraphs.

• How about proving the same thing for simple graphs?\(^a\)

• How to modify the proof to reduce 4SAT to MAX CUT?\(^b\)

• All NP-complete problems are mutually reducible by definition.\(^c\)
  – So they are equally hard in this sense.\(^d\)

---

\(^a\) Contributed by Mr. Tai-Dai Chou (J93922005) on June 2, 2005.

\(^b\) Contributed by Mr. Chien-Lin Chen (J94922015) on June 8, 2006.

\(^c\) Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.

\(^d\) Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.
MAX BISECTION

- MAX CUT becomes MAX BISECTION if we require that $|S| = |V - S|$.
- It has many applications, especially in VLSI layout.
MAX BISECTION Is NP-Complete

- We shall reduce the more general MAX CUT to MAX BISECTION.
- Add $|V| = n$ isolated nodes to $G$ to yield $G'$.
- $G'$ has $2n$ nodes.
- $G''$’s goal $K$ is identical to $G$’s
  - As the new nodes have no edges, they contribute 0 to the cut.
- This completes the reduction.
The Proof (concluded)

- Every cut \((S, V - S)\) of \(G = (V, E)\) can be made into a bisection by appropriately allocating the new nodes between \(S\) and \(V - S\).

- Hence each cut of \(G\) can be made a cut of \(G'\) of the same size, and vice versa.
BISECTION WIDTH

- BISECTION WIDTH is like MAX BISECTION except that it asks if there is a bisection of size at most $K$ (sort of MIN BISECTION).

- Unlike MIN CUT, BISECTION WIDTH is NP-complete.

- We reduce MAX BISECTION to BISECTION WIDTH.

- Given a graph $G = (V, E)$, where $|V|$ is even, we generate the complement\(^a\) of $G$.

- Given a goal of $K$, we generate a goal of $n^2 - K$.\(^b\)

---

\(^a\)Recall p. 379.

\(^b\)|$V$| = $2n$. 
The Proof (concluded)

- To show the reduction works, simply notice the following easily verifiable claims.
  - A graph $G = (V, E)$, where $|V| = 2n$, has a bisection of size $K$ if and only if the complement of $G$ has a bisection of size $n^2 - K$.
  - So $G$ has a bisection of size $\geq K$ if and only if its complement has a bisection of size $\leq n^2 - K$. 
HAMILTONIAN PATH Is NP-Complete\textsuperscript{a}

Theorem 47  Given an undirected graph, the question whether it has a Hamiltonian path is NP-complete.

\textsuperscript{a}Karp (1972).
A Hamiltonian Path at IKEA, Covina, California?
TSP (D) Is NP-Complete

Corollary 48 TSP (D) is NP-complete.

• Consider a graph $G$ with $n$ nodes.

• Create a weighted complete graph $G'$ with the same nodes as $G$.

• Set $d_{ij} = 1$ on $G'$ if $[i, j] \in G$ and $d_{ij} = 2$ on $G'$ if $[i, j] \notin G$.
  
  – Note that $G'$ is a complete graph.

• Set the budget $B = n + 1$.

• This completes the reduction.
TSP (D) Is NP-Complete (continued)

- Suppose $G'$ has a tour of distance at most $n + 1$.\(^a\)
- Then that tour on $G'$ must contain at most one edge with weight 2.
- If a tour on $G'$ contains one edge with weight 2, remove that edge to arrive at a Hamiltonian path for $G$.
- Suppose a tour on $G'$ contains no edge with weight 2.
- Remove any edge to arrive at a Hamiltonian path for $G$.

\(^a\)A tour is a cycle, not a path.
TSP (D) Is NP-Complete (concluded)

- On the other hand, suppose $G$ has a Hamiltonian path.
- There is a tour on $G'$ containing at most one edge with weight 2.
  - Start with a Hamiltonian path and then close the loop.
- The total cost is then at most $(n - 1) + 2 = n + 1 = B$.
- We conclude that there is a tour of length $B$ or less on $G'$ if and only if $G$ has a Hamiltonian path.
Random TSP

- Suppose each distance $d_{ij}$ is picked uniformly and independently from the interval $[0, 1]$.
- It is known that the total distance of the shortest tour has a mean value of $\beta \sqrt{n}$ for some positive $\beta$.\textsuperscript{a}
- In fact, the total distance of the shortest tour deviates from the mean by more than $t$ with probability at most $e^{-t^2/(4n)!}$\textsuperscript{b}

\textsuperscript{a}Beardwood, Halton, & Hammersley (1959).
\textsuperscript{b}Dubhashi & Panconesi (2012).
Graph Coloring

• $k$-COLORING: Can the nodes of a graph be colored with $\leq k$ colors such that no two adjacent nodes have the same color?\(^a\)

• 2-COLORING is in P (why?).

• But 3-COLORING is NP-complete (see next page).

• $k$-COLORING is NP-complete for $k \geq 3$ (why?).

• EXACT-$k$-COLORING asks if the nodes of a graph can be colored using *exactly* $k$ colors.

• It remains NP-complete for $k \geq 3$ (why?).

\(^a\) is *not* part of the input; $k$ is part of the problem statement.
3-COLORING is NP-Complete\textsuperscript{a}

- We will reduce NAESAT to 3-COLORING.
- We are given a set of clauses $C_1, C_2, \ldots, C_m$ each with 3 literals.
- The boolean variables are $x_1, x_2, \ldots, x_n$.
- We shall construct a graph $G$ that can be colored with colors $\{0, 1, 2\}$ if and only if all the clauses can be NAE-satisfied.

\textsuperscript{a}Karp (1972).
The Proof (continued)

• Every variable $x_i$ is involved in a triangle $[a, x_i, \neg x_i]$ with a common node $a$.

• Each clause $C_i = (c_{i1} \lor c_{i2} \lor c_{i3})$ is also represented by a triangle
  
  $[c_{i1}, c_{i2}, c_{i3}]$.

  – Node $c_{ij}$ and a node in an $a$-triangle $[a, x_k, \neg x_k]$ with the same label represent distinct nodes.

• There is an edge between $c_{ij}$ and the node that represents the $j$th literal of $C_i$.

---

\textsuperscript{a}Alternative proof: There is an edge between $\neg c_{ij}$ and the node that represents the $j$th literal of $C_i$. Contributed by Mr. Ren-Shuo Liu (D98922016) on October 27, 2009.
Construction for $\cdots \land (x_1 \lor \neg x_2 \lor \neg x_3) \land \cdots$

![Diagram](image)
The Proof (continued)

Suppose the graph is 3-colorable.

- Assume without loss of generality that node $a$ takes the color 2.
- A triangle must use up all 3 colors.
- As a result, one of $x_i$ and $\neg x_i$ must take the color 0 and the other 1.
The Proof (continued)

• Treat 1 as \texttt{true} and 0 as \texttt{false}.\textsuperscript{a}
  
  – We are dealing with the \texttt{a}-triangles here, not the clause triangles yet.

• The resulting truth assignment is clearly contradiction free.

• As each clause triangle contains one color 1 and one color 0, the clauses are \texttt{NAE}-satisfied.

\textsuperscript{a}The opposite also works.
The Proof (continued)

Suppose the clauses are NAE-satisfiable.

- Color node $a$ with color 2.
- Color the nodes representing literals by their truth values (color 0 for false and color 1 for true).
  - We are dealing with the $a$-triangles here, not the clause triangles.
The Proof (continued)

• For each clause triangle:
  – Pick any two literals with opposite truth values.\(^a\)
  – Color the corresponding nodes with 0 if the literal is \textit{true} and 1 if it is \textit{false}.
  – Color the remaining node with color 2.

\(^a\)Break ties arbitrarily.
The Proof (concluded)

- The coloring is legitimate.
  - If literal $w$ of a clause triangle has color 2, then its color will never be an issue.
  - If literal $w$ of a clause triangle has color 1, then it must be connected up to literal $w$ with color 0.
  - If literal $w$ of a clause triangle has color 0, then it must be connected up to literal $w$ with color 1.
More on 3-COLORING and the Chromatic Number

- 3-COLORING remains NP-complete for planar graphs.\(^a\)

- Assume \(G\) is 3-colorable.

- There is a classic algorithm that finds a 3-coloring in time \(O(3^{n/3}) = 1.4422^n\).\(^b\)

- It can be improved to \(O(1.3289^n)\).\(^c\)

\(^a\)Garey, Johnson, & Stockmeyer (1976); Dailey (1980).
\(^b\)Lawler (1976).
\(^c\)Beigel & Eppstein (2000).
More on 3-COLORING and the Chromatic Number (concluded)

- The **chromatic number** $\chi(G)$ is the smallest number of colors needed to color a graph $G$.

- There is an algorithm to find $\chi(G)$ in time $O((4/3)^{n/3}) = 2.4422^n$.

- It can be improved to $O((4/3 + 3^{4/3}/4)^n) = O(2.4150^n)$ and $2^n n^{O(1)}$.

- Computing $\chi(G)$ cannot be easier than 3-COLORING.

---

\(^a\) Lawler (1976).
\(^b\) Eppstein (2003).
\(^c\) Koivisto (2006).
\(^d\) Contributed by Mr. Ching-Hua Yu (D00921025) on October 30, 2012.