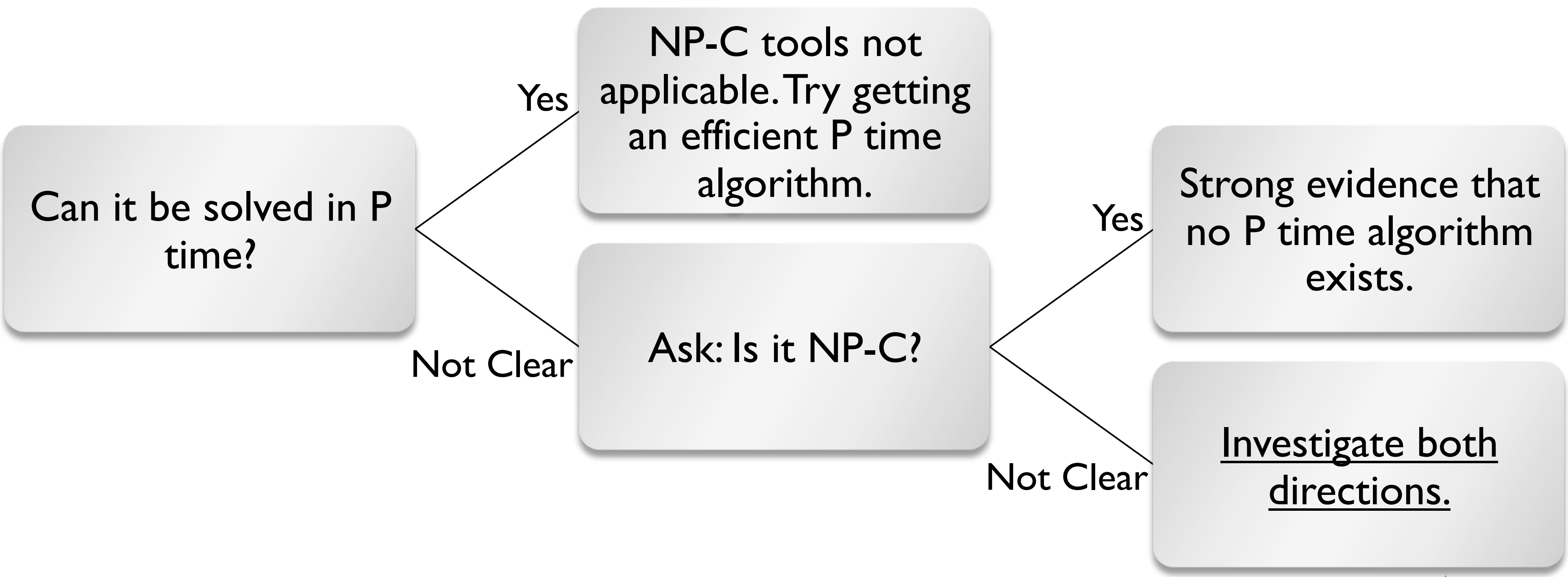
Using NP-Completeness for Problem Analysis

Theja Tulabandhula
Oct 7 2011

Given a new decision problem (say in NP)



a. Failed NP-C proof might help getting P time algorithm (vice-versa)

b. Caution: If P ≠ NP, there are problems not in P or NP-C.

Intuition is not trustworthy

- Intuition based on related problems.
- Many problems that are P time only differ slightly from NP-C counterparts.

P	NP-C
2-SAT	3-SAT
2DM	3DM
Shortest path between two vertices	Longest path between two vertices
Edge cover	Vertex cover
Transitive reduction	Minimum equivalence digraph
Intree scheduling	Outtree scheduling

Post NP-C proof: Continuing two-sided Analysis

- Investigating sub-problems of a problem.
  - Motivated by re-adding details we left out
  - Relevant cases might be P time
  - Only small set of instances may have made it NP-C.

- Map the boundary between P time and NP-C.
  - Illustration via GRAPH 3-COLORABILITY
- Special sub-problems where numbers are important: Pseudo-Polynomial time

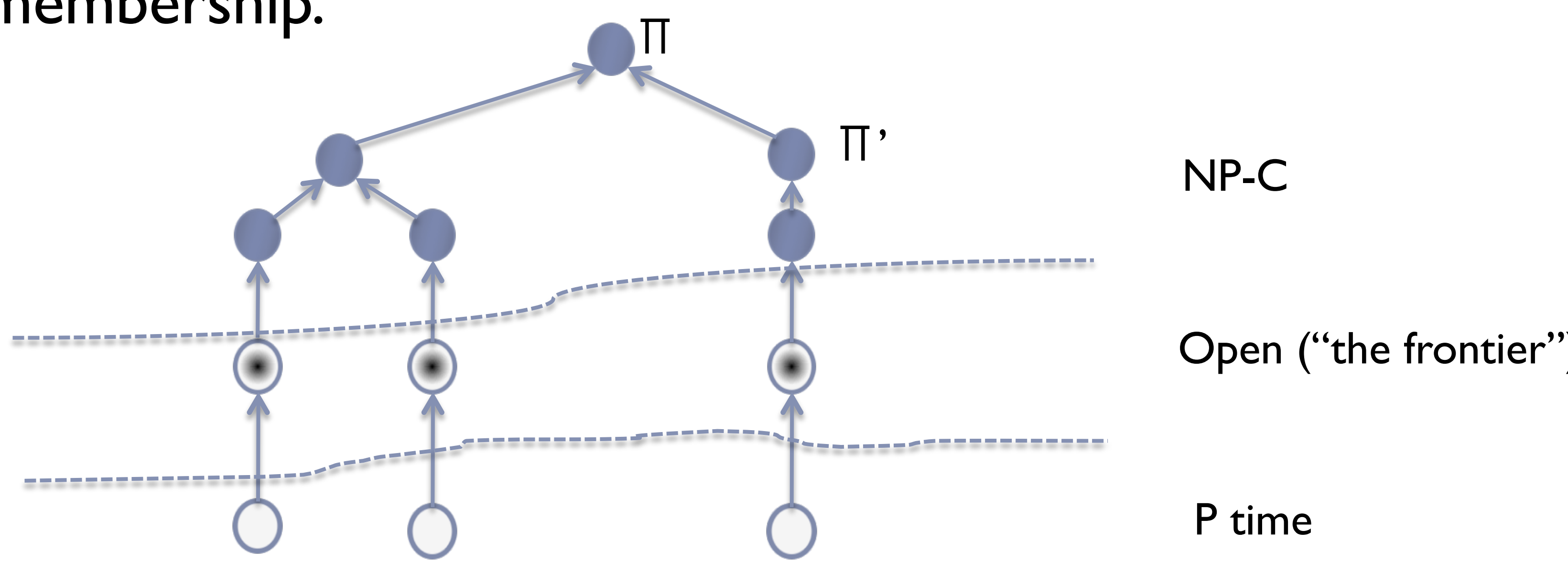
Analyzing sub-problems

- (Π, D, Y) = NP-C problem, {all instances}, {yes instances}.
- Subproblem (Π', D', Y') such that $D' \subset D$ and $Y' = Y \cap D$

- Examples
  - restrictions like planar, acyclic, bipartite on graphs.
  - $|A| \leq N$ for sets.

- Pick sub-problems
  - depending on application or
  - if they seem natural

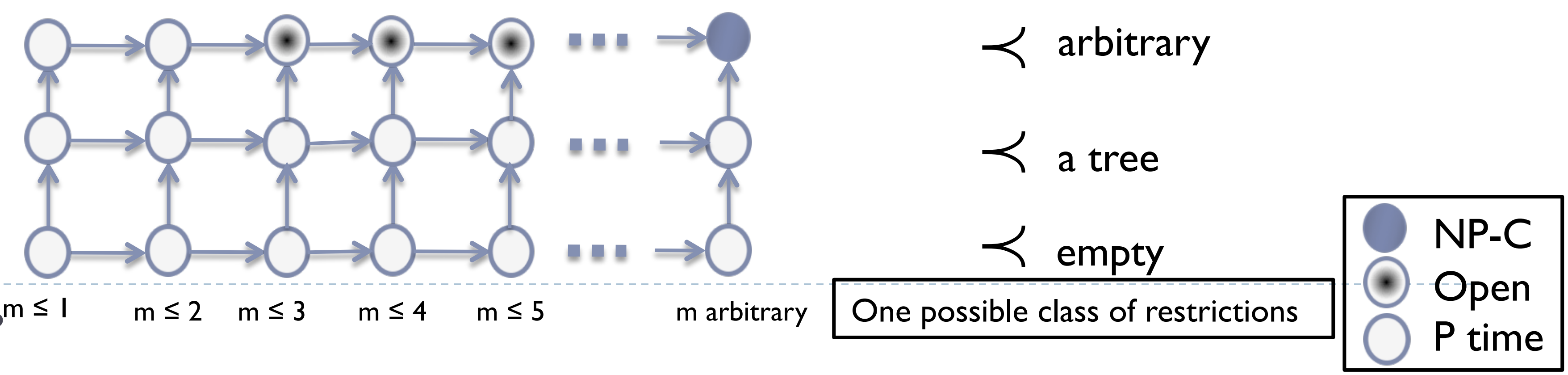
- Each Π' may independently be NP-C or P or unknown.
  - Example: Π = SATISFIABILITY
    - Π' 3-SAT is NP-C
    - Π'' 2-SAT is P.
- Goal: Analyze sub-problems and determine their membership.



Concrete Example: Precedence constrained scheduling Next.

Scheduling equal length tasks subject to partial order.

- Given
  - set T = {t1, t2,…} with equal length tasks.
  - Partial order $\prec$ on T
  - m processors
  - One overall deadline.
- Question: Is there a schedule s.t.
  - each period, at most m tasks are scheduled
  - if t $\prec$ t', then schedule t before t'.



Minimal NP-C and Maximal P time

- [Def] Given C = {Π', Π'',…} of sub-problems of an NP-C problem
  - Π' in C is a minimal NP-C subproblem if
    - it is NP-C
    - no sub-problem of Π' also in C is known to be NP-C
  - Π'' in C is a maximal P time subproblem if
    - it is P time
    - no problem in C containing Π'' as a subproblem is known to be P

- Example:
  - {$\prec$ arbitrary, $m \leq 2$} is maximal P time subproblem
  - {$\prec$ arbitrary, m arbitrary} is minimal NP-C subproblem

Illustration : GRAPH 3-COLORABILITY

- Given
  - G(V,E)
- Question: Is G 3-colorable?
  - does there exist an assignment of colors to all vertices
  - s.t. color(v) ≠ color(u) for all vertices v,u in V.

- Related to 4-color conjecture, scheduling, partitioning
- Sub-problem of K-COLORABILITY
- NP-C (Stockmeyer '73)

- First restriction we consider: degree boundedness.

A note on Degree boundedness

- Many graph problems are in P if degree bound sufficiently small (table).

	P time for d $\leq$	NP-C for d $\geq$
VERTEX COVER	2	3
HAMILTONIAN CIRCUIT	2	3
3-COLORABILITY	3	4
FEEDBACK VERTEX SET	2	3

Prove from general problem using Local Replacement

<u>KEY IDEA</u>

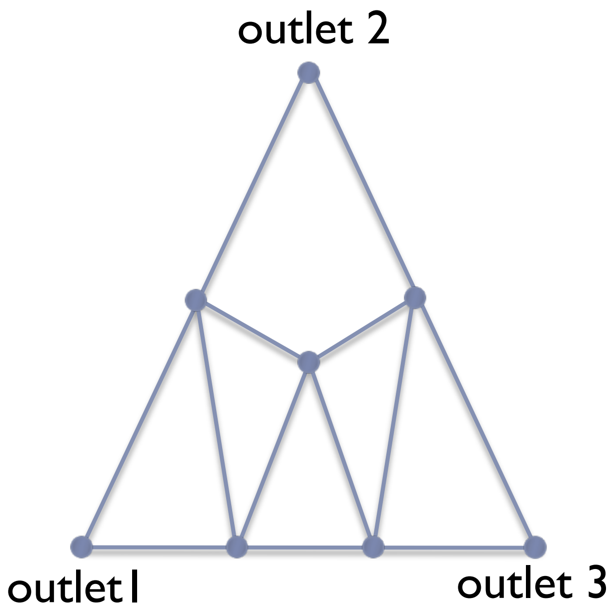
Vertex Substitute

- CLIQUE is in P for any fixed degree bound d.
  - For fixed d, search over all d subsets in P time.

Thm: 3-COLORABILITY with $d \leq 4$ is NP-C

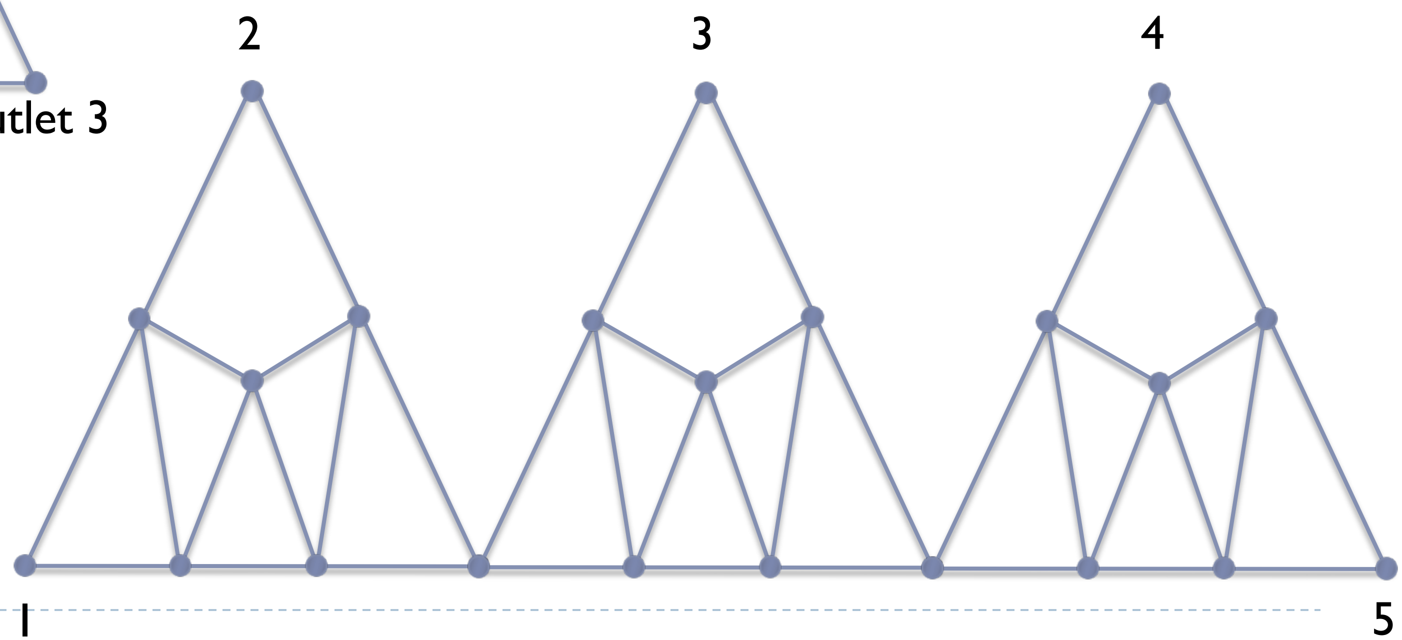
- Proof Step 1:
  - NP membership follows from general problem (d arbitrary).
- Proof Step 2:
  - We will transform the general problem into an instance of problem with $d \leq 4$.
  - Let G = (V,E) be general instance.
  - Construct corresponding G' = (V',E') with $d \leq 4$ such that
  - G' is 3-colorable if and only if G is 3-colorable.

Vertex Substitute gadget: Local replacement technique

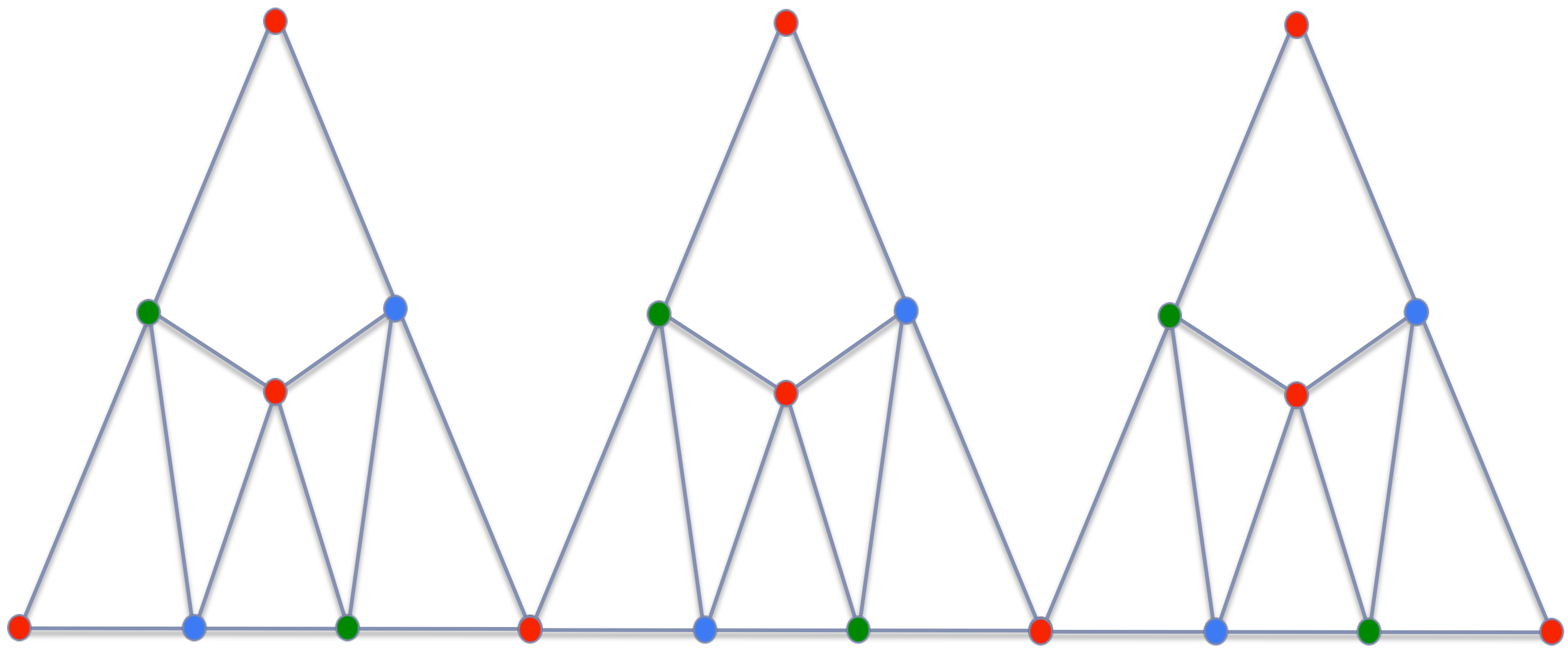


H$_3$
8 Vertices
3 Outlets (d = 2)

H$_5$ (H$_k$)
Adjoining H$_3$ to H$_4$ (H$_{k-1}$)
1st outlet of H$_3$ = 4th (k-1)th outlet of H$_4$ (H$_{k-1}$)
Rename outlets 4 and 5 (k-1 and k)



Properties of the gadget.



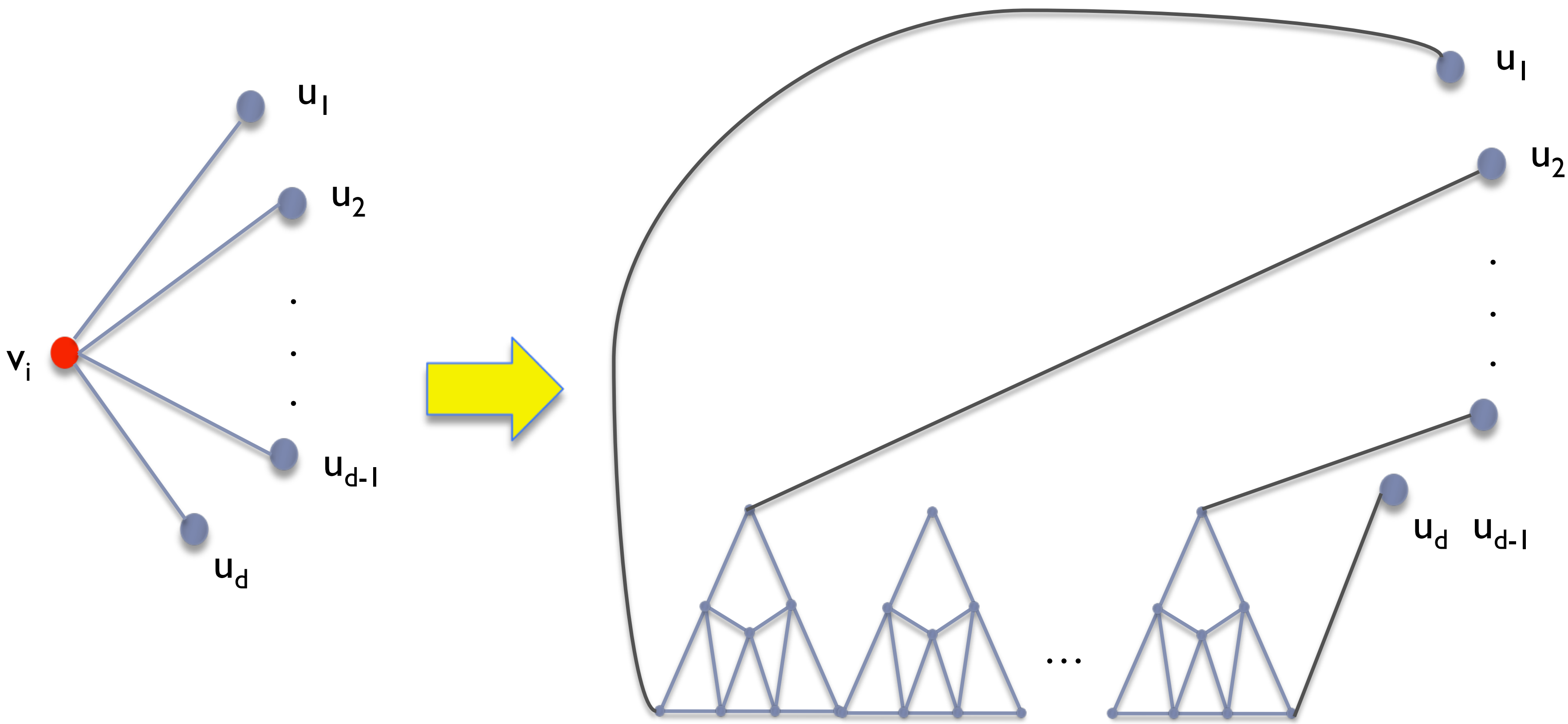
Proof Step 2 (contd.)

- Properties of Vertex Substitute gadget Hk
  - H_k has k labeled outlets and 7(k-2)+1 vertices
  - max degree of H_k = 4
  - Outlet degree = 2
  - H_k is 3-colorable but NOT 2-colorable.
  - 3-coloring of H_k : all outlets have same color.

- Let $v_1, \ldots, v_r$ be the r vertices of G with d > 4
- Plan: Construct a sequence of graphs $G_0, G_1, \ldots, G_r$ such that
  - $G_0 = G$
  - $G_r = G'$

G_0 G_r

- G_i is constructed from G_{i-1}
- Let v_i in G_{i-1} have deg d
- Delete v_i and insert H_d



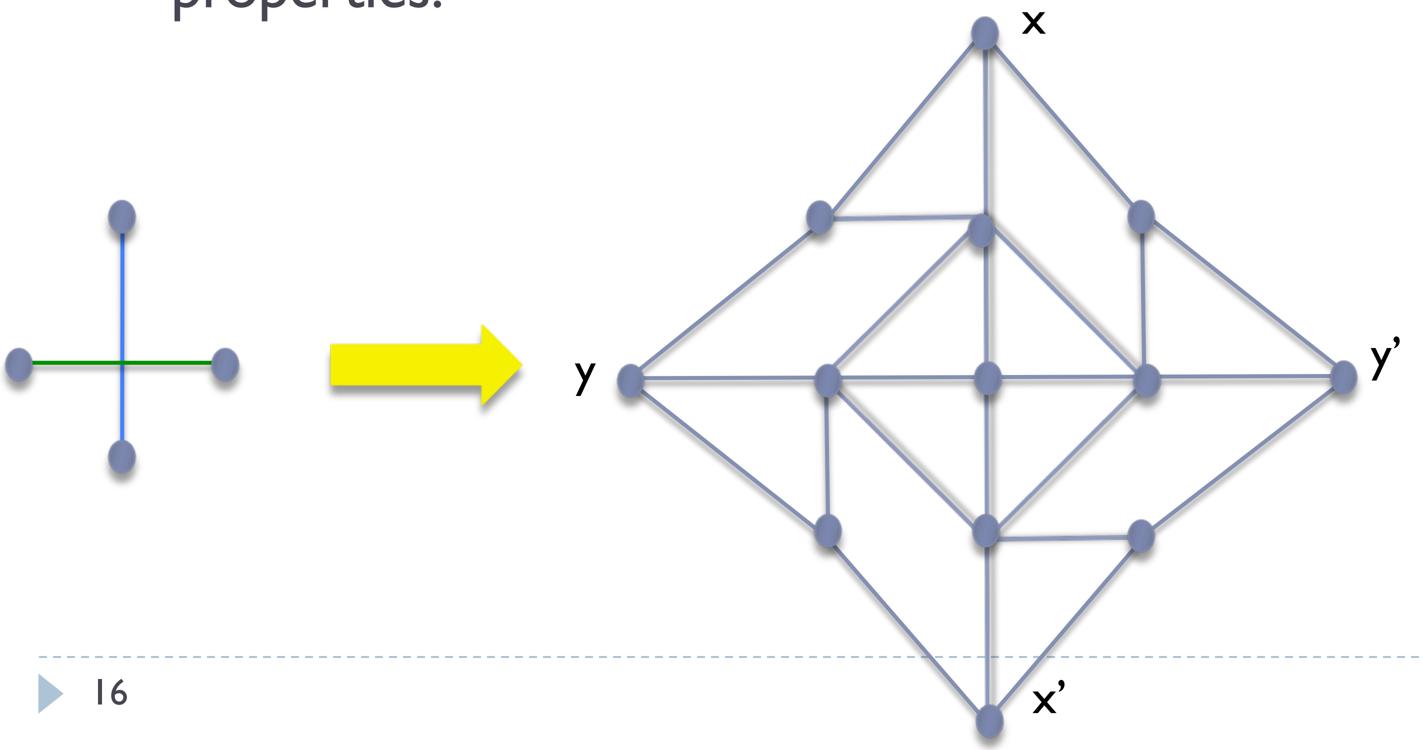
- For each k, G_k has r-k vertices with degree exceeding 4.
- G_k is 3-colorable iff G_0 (=G) is 3-colorable.

Next Restriction: Planarity

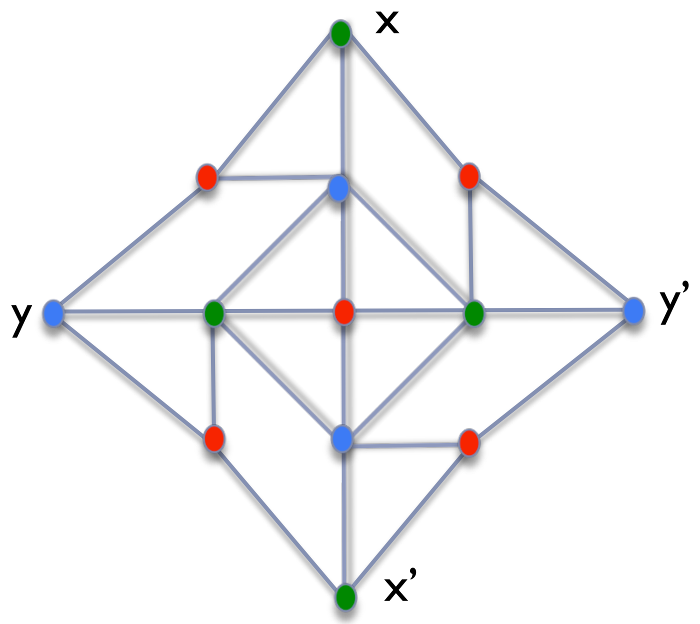
- G is Planar if it can be embedded in the plane.
- No two edges intersect except at a common endpoint.
- Example 1: CLIQUE
  - A planar graph is K_5 free.
- Example 2: MAX CUT
  - Given G and positive edge weights and positive integer K
  - Question: Can we split V into V_1,V_2 such that Σ [weights of edges with one endpoint in each set] $\geq$ K?
  - General problem NP-C
  - Subproblem with weights all equal is NP-C
  - Subproblem with planarity is in P time!
- Usual Trick: Local Replacement OR Planarity Preserving Transform

Thm: Planar 3-COLORABILITY is NP-C

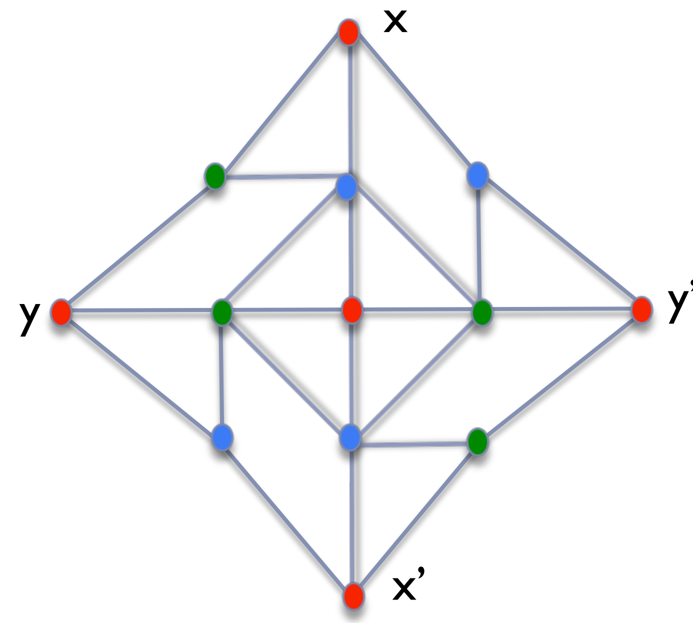
- Proof Step 1: Belongs to NP
- Proof Step 2: (Design a "crossover" : use in place of edge crossings)
  - Construct G' such that G' is 3-colorable iff G is.
  - "Crossover" graph H with "outlets" x, x', y, y' has two properties.



- Property 1: Any 3-coloring of H => f(x) = f(x') AND f(y)=f(y')
- Property 2: There exist 3-colorings f and g such that
  - f(x) = f(x') = f(y) = f(y')
  - g(x) = g(x') ≠ g(y) = g(y')



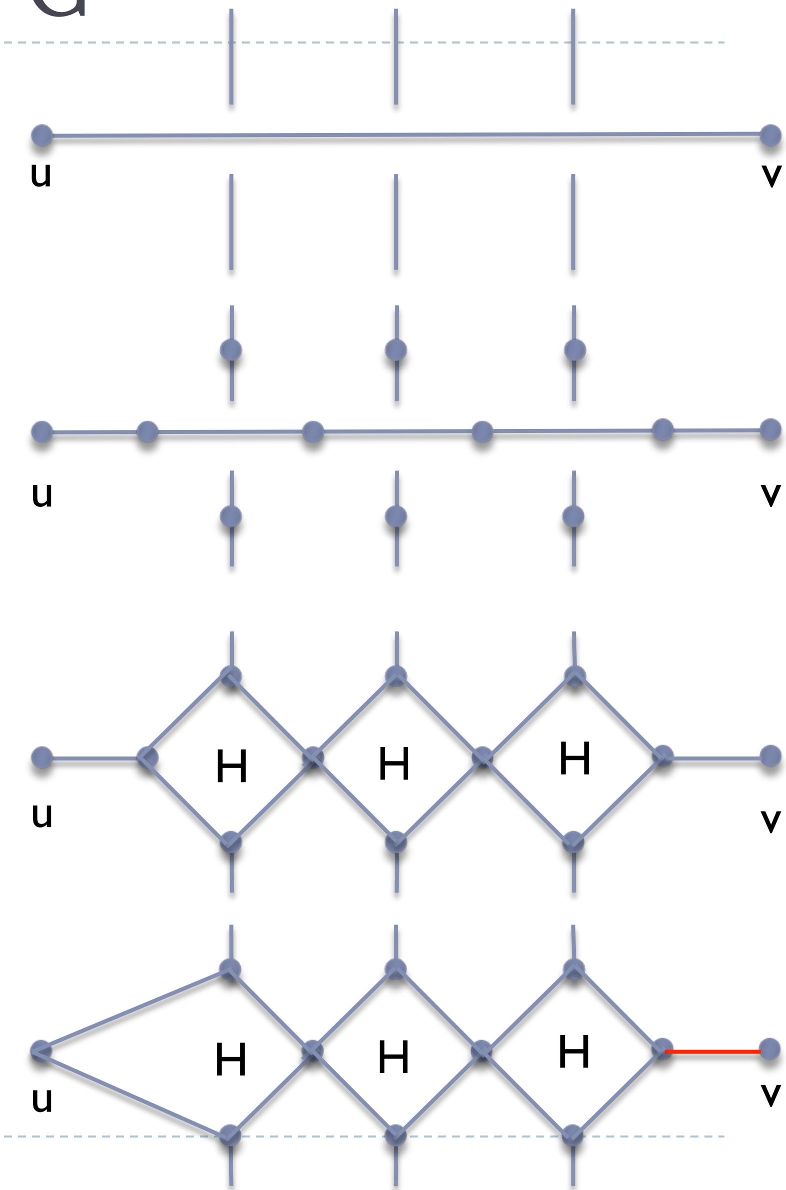
satisfies Property 1
satisfies Property 2 (different colors)



satisfies Property 1
satisfies Property 2 (same colors)

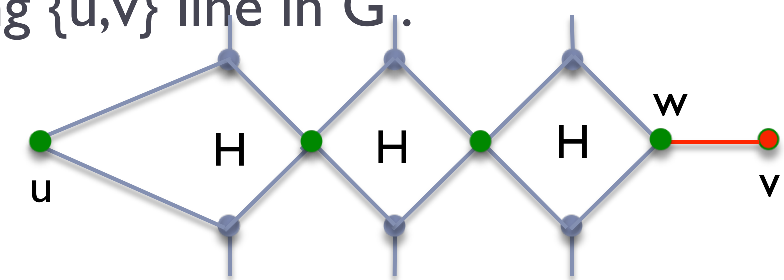
Construction of G' from G

- Embed G in plane, allow crossings : P time
- Pick {u,v}-line which has crossings.
- Add vertices.
- Replace each crossing with H
- Coalesce u with its nearest *new* point.
- Define operant edge: between v and its nearest new point.



Showing that G' is 3-colorable iff G is.

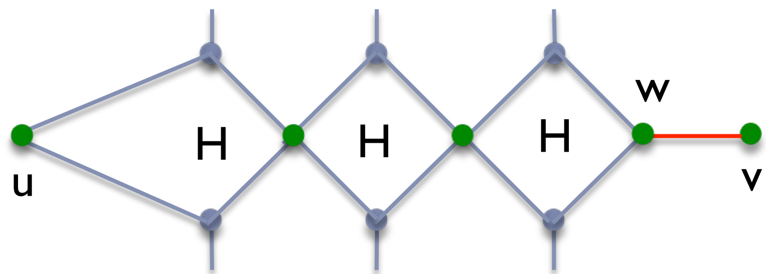
- => (direct proof)
  - f is any 3-coloring of G'. Claim: f|V is a 3-coloring of G.
  - Case 1: There is a {u,v} in G without any crossings. Then f(u) ≠ f (v) in both graphs.
  - Case 2: There is a {u,v} in G which had crossings. Consider the corresponding {u,v} line in G'.



  - All new points on {u,v} line should have same color as u.
  - Operant edge {w,v} end points have different color.
  - Thus f|V is a 3-coloring.

Showing that G' is 3-colorable iff G is.

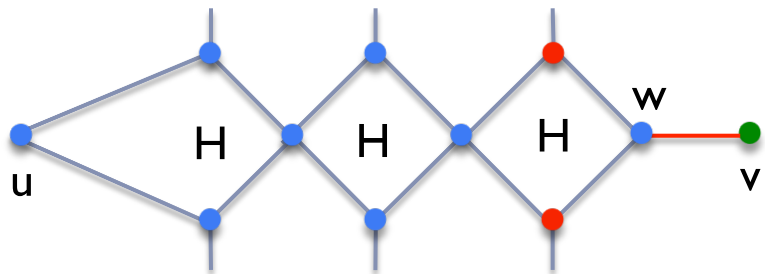
- => (Contrapositive proof)
  - f is any 3-coloring of G'. Claim: f|V is a 3-coloring of G.
  - Assume otherwise. There is a {u,v} in G such that f(u) = f(v)
  - Consider {u,v} line in G'.



  - All new points on {u,v} should have same color as u.
  - => Operant edge {w,v} end points have same color.
  - Thus, if f|V is not a 3-coloring of G, then f is not a 3-coloring of G'.

Showing that G' is 3-colorable iff G is.

- <=
  - Let f be any 3-coloring of G. Extend to G' as follows:
  - Color new points on {u,v} with f(u).



  - => f(w) ≠ f(v)
  - By Property 2, this partial coloring of G' extends to
    - internal vertices
    - crossovers

Number Problems

- Analyzing subproblems critical with number problems
- Example: PARTITION
  - Given: Set $A = \{a1, a2, \ldots, an\}$ and positive numbers $\{s(a1), s(a2), \ldots, s(an)\}$ and $B = \sum_i s(ai)$
  - Question: Is there a subset A' such that $\sum_{A'} s(ai) = \sum_{A \setminus A'} s(ai)$
- Eg: 4 elements: $s(a1) = 1$, $s(a2) = 3$, $s(a3)=6$, $s(a4)=2$. $B = 12$.
- Table: Assign cell (I,j) T if some subset with items {a1,..,ai} has $\sum_{k \le i} s(ak) = j$.

(I,j)	0	1	2	3	4	5	B/2 = 6
item 1	T	T	F	F	F	F	F
item 2	T	T	F	F	F	F	F
item 3	T	T	F	T	T	F	T
item 4	T	T	T	T	T	T	T

Pseudo-polynomial time

- Table method is polynomial in nB
- But instance length is O(nlogB) (conciseness of encoding)
- nB is not bounded by any polynomial of nlogB
- Thus, Table method is NOT a P time algo for PARTITION.

- We see that NP-C of PARTITION depends STRONGLY on the fact that large input numbers are allowed.
- <u>Pseudo-polynomial time algorithms</u>: If numbers upper bounded by O(Length[I]), then P time
- Can be useful and practical.

Summary

- Instances from your application will often satisfy special constraints affecting NP-C membership.
- We saw two restrictions of 3-COLORABILITY
- Used our "Local Replacement" technique
- Interaction between numbers and NP-C