Non-Approximability Results
(2nd part continued)

Summary

- The PCP theorem
  - Application: non-approximability of MAXIMUM CLIQUE
- Input-Dependent and Asymptotic Approximation
  - Approximation algorithm for graph colouring
  - Approximation algorithm for set cover
  - Asymptotic approximation scheme for edge colouring

MAXIMUM CLIQUE

- INSTANCE: Graph $G=(V,E)$
- SOLUTION: A subset U of V such that, for any two vertices u and v in U, (u,v) is in E
- MEASURE: Cardinality of U

Inapproximability of clique

Theorem: MAX CLIQUE $\notin$ PTAS unless P=NP

Proof:

- We show that it's possible to "reduce" MAX 3-SAT to MAX CLIQUE preserving the approximation for all *r*. Given that MAX 3-SAT $\notin$ PTAS, the thesis follows.
- Let (U, C) an instance of MAX 3-SAT. U={variables} C={clauses}
- Define the MAX 3-SAT instance as: G=(V, E)
  - $V = \{ (l, c) \mid l \in c \wedge c \in C \}$, $E = \{ ((l_1, c_1), (l_2, c_2)) \mid l_1 \neq \neg l_2 \wedge c_1 \neq c_2 \}$
  - l_i are literals and c_i are clauses

Inapproximability of clique

Proof (continued):

  - $V = \{ (l, c) \mid l \in c \wedge c \in C \}$, $E = \{((l_1, c_1), (l_2, c_2)) \mid l_1 \neq \neg l_2 \wedge c_1 \neq c_2\}$
- For any clique V', let f the truth assignment as follows:
  - f(u) = true iff exists a clause c such that $(u, c) \in V'$.
- It's easy to show that f() is a consistent truth assignment.
- From E, f() can satisfies $\geq |V'|$ clauses: $m((U, C), f) \geq |V'|$
- It's easy to show that the max number of satisfiable clauses is equal to the size of the max clique in G.
  - Given a set of clauses $C' \subseteq C$, for any truth assignment f' for C' and for any $c \in C'$, let l_c any literal of c with $f'(l_c)$=true. The set of nodes (l_c, c) is clearly a clique in G.

Inapproximability of clique

Proof (continued):

- Therefore, any polynomial-time approximation scheme for MAX CLIQUE can be transformed in a polynomial-time approximation scheme for MAX 3-SAT.
- But, unless P=NP, MAX 3-SAT $\notin$ PTAS, so the thesis.

Inapproximability of clique

- MAX CLIQUE has a particular property, *self-improvability,* that yields the following result
- Theorem: MAX CLIQUE $\notin$ APX unless P=NP
- Proof:
  - If there exists an polynomial-time r'-approximation algorithm *A* for MAX CLIQUE, given an instance G, we will transform G into another, larger, instance f(G) and apply *A* to f(G).
  - The approximate solution *A*(f(G)) can be used to find a better approximate solution to G... therefore *A* can be transformed to an approximation scheme.

Inapproximability of clique

- The *self-improvability property:* Product graphs
  - Given a graph $G=(V,E)$, define $G^k(V^k,E^k)$ as
    $V^k=\{(v_1,v_2,\ldots,v_k) \mid v_i \in V\}$ (k-th Cartesian product of V)
    $E^k=\{(\boldsymbol{u},\boldsymbol{v}) \mid (u_i=v_i) \vee (u_i,v_i) \in E \textit{ for all } i\}$
  - If $C \subseteq V$ is a clique in G, it is easy to verify that $\{(v_1,v_2,\ldots,v_k) \mid v_i \in C \text{ for every } i\}$ is a clique in f(G) of size $|C|^k$
    - $m^*(f(G)) \geq (m^*(G))^k$
  - If $C' \subseteq V^k$ is a clique in f(G) with m^k verticies, then at least a coordinate *i* of the vertices $(v_1,v_2,\ldots,v_k)$ where there are m different vertices v_i in C'. These vertices are a clique in G of size $|C'|^{1/k.}$. Let *g* the procedure that builds this clique from C'.

Inapproximability of clique

- With *A* we can determine that

$$m^*(G) / m(G, g(A(f(G)))) \leq (m^*(f(G)) / m(f(G), A(f(G))))^{1/k}$$
$$\leq r'^{1/k}$$

- For any r>1, choosing $k \geq \log r'/\log r$, we obtain a polynomial-time approximation scheme for MAX CLIQUE.
- Last theorem states that it's impossible unless P=NP

The NPO world if P≠NP

NPO	MINIMUM TSP MAXIMUM CLIQUE
APX	MINIMUM BIN PACKING MAXIMUM SAT MINIMUM VERTEX COVER(⇩ ?) MAXIMUM CUT(⇩ ?)
PTAS	MINIMUM PARTITION
PO	MINIMUM PATH

MINIMUM GRAPH COLORING? Certainly not in PTAS

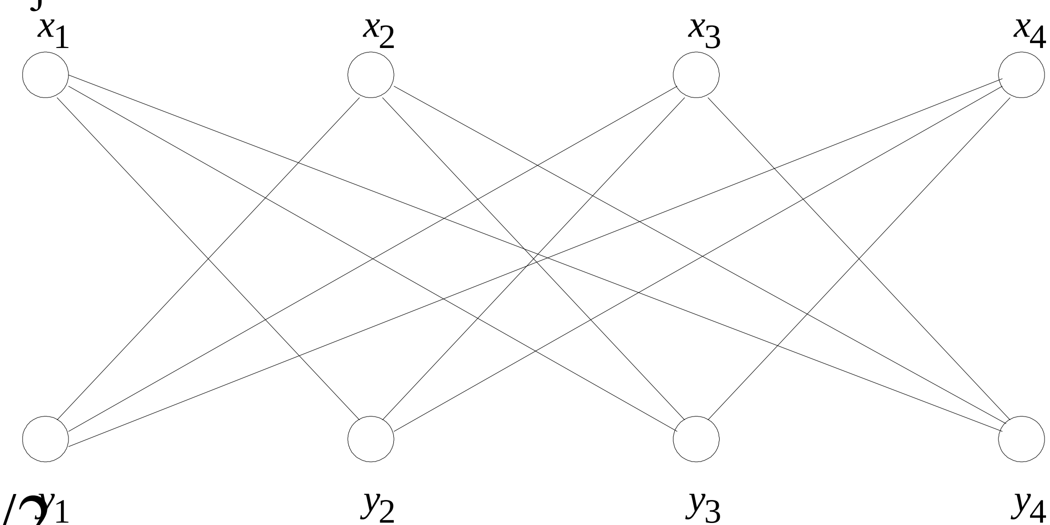
Input-Dependent Approximation

- A sequential algorithm for MINIMUM GRAPH COLORING

```
begin
    sort V in decreasing order with respect to the degree;
    for each node v do
        if there exists color not used by neighbors of v then assign this color to v
        else create new color and assign it to v
end.
```

Input-Dependent Approximation

- Performance of sequential algorithm
  - $G=(\{x_1,\ldots x_n, y_1,\ldots y_n\}, \{\{x_i,y_j\} \mid i\neq j\})$
  - $d(x_i)=d(y_j)=n-1$
  - The order $(x_1,y_1\ldots x_n,y_n)$ requires n colours
  - The optimal value is 2
  - The performance ratio is n/2
  - Generalizing, the performance ratio is $\Delta+1$ where Δ is the highest degree of nodes in G



Input-Dependent Approximation

- Polynomial-time $n/\log n$-approximation algorithm for MINIMUM GRAPH COLORING

```
begin
        i:=0; U:=V;
        while U ≠ ø do
        begin
                i:=i+1;V[i]:=ø;W:=U;H:=graph induced by W;
                while W ≠ ø do
                begin
                        v=node of minimum degree in H;
                        insert v in V[i];
                        delete v and its neighbours from W;
                        U:=U-V[i]
                end
        end
end.
```

Input-Dependent Approximation

We first prove that, if G is k-colorable, then the algorithm uses at most $3|V|/\log_k|V|$ colours

- At any iteration of the inner loop, H is k-colorable
- Hence, it contains an independent set of at least $|W|/k$ nodes of degree at most $|W|(k-1)/k$
- Minimum degree in H is at most $|W|(k-1)/k$
- At least $|W|-|W|(k-1)/k = |W|/k$ nodes will be in W at the next iteration (after remove the $|W|/k$ IS)
- Inner loop ends when W is empty
  - At least $\log_k|W|$ iterations are necessary

Input-Dependent Approximation

- At the end of inner loop
  - $|\{v \mid v \text{ in } W \text{ and } V[v] = i\}| \geq \log_k|W|$
- For each colour i , the number of vertices coloured with i is at least $\log_k|U|$, where U is the set of uncoloured nodes before the color i is used
- Before the first outer loop, if $|U| \geq |V|/\log_k|V|$, $\log_k|U| \geq \log_k(|V|/\log_k|V|) \geq ½ \log_k|V|$
  - U size decrease by at least $½ \log_k|V|$ at each iteration
  - The first time $|U|$ becomes smaller than $|V|/\log_k|V|$, the algorithm has used no more than $2\ |V|/\log_k|V|$ colours

Input-Dependent Approximation

- if $|U| < |V|/\log_k|V|$, to colour the remaining nodes $|V|/\log_k|V|$ colours suffice
- That is, the algorithm uses at most $3|V|/\log_k|V|$ colours
- The algorithm uses at most $3|V|/\log_{m^*(G)}|V|$, that is, at most $3n \log(m^*(G))/\log n$ colours
- The performance ratio is at most $(3n \log(m^*(G))/\log n) / m^*(G) = O(n / \log n)$

MINIMUM SET COVER

- INSTANCE: Collection C of subsets of a finite set S

- SOLUTION: A set cover for S, i.e., a subset C' of C such that every element in S belongs to at least one member of C'

- MEASURE: $|C'|$

Input-Dependent Approximation

- Johnson's algorithm
  - Polynomial-time logarithmic approximation algorithm for MINIMUM SET COVER

```
begin
    U:=S;
    for any set c_i in C do c'_i := c_i ;
    C':=ø;
    repeat
        i:=index of c'with maximum cardinality;
        insert c_i in C';
        U := U-{elements of c'_i };
        delete all elements of c_i from all c';
    until U:=ø
end.
```

Input-Dependent Approximation

- It is possible to show that Johnson's algorithm is a (ln n + 1)-approximate algorithm for the MINIMUM SET COVER, where n is the number of elements of S

Class F-APX

- Let F be a class of functions

- The class F -APX contains all NPO problems P that admit a polynomial-time algorithm A such that, for any instance x of P, $R(x, A(x)) \leq f(|x|)$, for a given function $f \in$ F

- P is said to be $f(n)$-approximable
- A is said to be an $f(n)$-approximation algorithm

Class APTAS

- The class APTAS contains all NPO problems P that admit a polynomial-time algorithm A and a constant k such that, for any instance x of P and for any rational r, $R(x, \mathsf{A}(x,r))) \leq r+k/m^*(x)$

- The time complexity of A is polynomial in $|x|$ but not necessarily in $1/(r-1)$
- A is said to be an *asymptotic approximation scheme*
  - A is clearly a $(r+k)$-approximation algorithm

The NPO world

NPO	
$O(n)$-APX	MINIMUM GRAPH COLORING
$O(\log n)$-APX	MINIMUM SET COVER
APX	MAXIMUM SAT MINIMUM VERTEX COVER MAXIMUM CUT
APTAS	MINIMUM EDGE COLORING
PTAS	MINIMUM PARTITION
PO	MINIMUM PATH