Approximation Algorithms I

Definitions

Vertex cover
Set cover
Partition

\{ NP-complete problems \}
\{ NP-hard \}

Approximation Algos & Schemes

An algorithm for a problem of size $n$ has an approximation ratio $\rho(n)$ if for any input, algorithm produces a solution of cost $C$ such that

$$\max \left( \frac{C}{C_{\text{opt}}} , \frac{C_{\text{opt}}}{C} \right) \leq \rho(n)$$

Algorithm is an $\rho(n)$-approximation algorithm.

An approximation scheme takes as input $\epsilon > 0$ and for any fixed $\epsilon$, the scheme is a $(1+\epsilon)$-approximation algorithm.

Polynomial time approximation scheme (PTAS): polynomial in $n$

Fully PTAS: polynomial in $n$ and $\frac{1}{\epsilon}$

$O(n^{2/\epsilon})$ PTAS not FPTAS.

$O(n^{1/\epsilon^2})$ FPTAS
Vertex Cover

Undirected graph $G(V, E)$
Find a subset $V' \subseteq V$ such that if $(u, v)$ is an edge of $G$, then either $u \in V'$ or $v \in V'$ or both.
Find $V'$ so $|V'|$ is minimum.

Approx-Vertex-Cover

$C \leftarrow \emptyset$
$E' \leftarrow E$

while $E' \neq \emptyset$

Pick $(u, v) \in E$ arbitrarily
$C \leftarrow C \cup \{u, v\}$
Delete from $E'$ all edges incident on $u$ or $v$

Return C

Runs in poly time. Produces a vertex cover.
How close to optimal?
Example

Approx-Vertex-Cover could pick \((b, c), (e, f), (d, g)\)
\[ C = \{b, c, d, e, f, g\} \quad |C| = 6 \]
Optimal solution \(C_{opt} = \{b, d, e\} \quad |C_{opt}| = 3 \)

Approx-Vertex-Cover is a 2-approximation algorithm

Proof: Let \(A\) denote the edges that are picked.
Optimal cover \(C_{opt}\) must include at least one endpoint of each edge in \(A\) (and other edges).
No two edges in \(A\) share an endpoint.
\(|A|\) is a lower bound for \(|C_{opt}|, |C_{opt}| \geq |A|\)
Number of vertices in \(C = 2|A|\)
\[ |C| \leq 2 |C_{opt}| \]

\(\sqrt{\text{ }}\)
Given a set $X$ and a family of (possibly overlapping) subsets $S_1, S_2, \ldots, S_m \subseteq X$ such that
\[ \bigcup_{i=1}^{m} S_i = X, \] find $C \subseteq \{1, 2, \ldots, m\}$ such that $\bigcup_{i \in C} S_i = X$, while minimizing $|C|$. 

Approx_set_cover (on next page) selects $S_1, S_4, S_5, S_3$ in that order.

Optimal: $S_3, S_4, S_5$
Approx-Set-Cover

\[ C = \emptyset \]

While elements in \( X \) remain

Pick largest \( S_i \) : \( C = C \cup S_i \)

Remove all elements in \( S_i \) from \( X \) and other \( S_j \)

Return \( C \)

Poly time, returns a cover

Approx-Set-Cover is a \((\ln(n)+1)\)-approximation algo

Proof: Assume there is a cover \( C_{\text{opt}} \) \( |C_{\text{opt}}| = t \)

Let \( X_k \) be set of elements in iteration \( k \) \( (X_0 = X) \)

\( \forall k, X_k \) can be covered by \( t \) sets.

\( \Rightarrow \) one of them covers at least \( \frac{|X_k|}{t} \) elements.

\( \Rightarrow \) algo picks a set of (current) size \( \geq \frac{|X_k|}{t} \)

\( \Rightarrow \) \( \forall k \) \( |X_{k+1}| \leq (1 - \frac{1}{t})|X_k| \)

[More careful analysis (see CLRS, Thm 35) relates \( \ln(n) \) to harmonic numbers. \( t \) should shrink!]
Proof (contd.)

\[ \forall k, |X_{k+1}| \leq \left(1 - \frac{1}{e}\right) |X_k| \]

\[ \Rightarrow \forall k, |X_k| \leq \left(1 - \frac{1}{e}\right)^k n \]

\[ \leq e^{-k/t} \cdot n \]

Algorithm terminates when \( |X_k| < 1 \), i.e., \( |X_k| = 0 \) and cost = \( k \).

\[ e^{-k/t} \cdot n < 1 \]

\[ e^{k/t} > n \]

When \( \frac{k}{t} > \ln(n) \) and algorithm terminates.

So we have an \( (\ln(n)+1) \)-approximation algorithm.

\[ \times \]

Approximation ratio gets worse for larger problems.
Partition

Set $S$ of $n$ items with weights $s_1, \ldots, s_n$

Assume $s_1 \geq s_2 \geq \ldots \geq s_n$ WLOG

Partition into $A$ and $B$ to minimize

$$\max \left( \frac{\leq s_i}{\text{w}(A)}, \frac{\leq s_i}{\text{w}(B)} \right)$$

Define $2L = \sum_{i=1}^{n} s_i = \text{w}(S)$

Optimum solution $\Rightarrow L$.

Want a PTAS. Note: 2-approx algo trivial.

(FPTAS also exist)
Approx - Partition

Define \( m = \left\lceil \frac{1}{\epsilon} \right\rceil - 1 \) \( \epsilon \approx \frac{1}{m+1} \)

First phase: Find an optimal partition \( A', B' \)
of \( S_1, \ldots, S_m \)

Second phase: \( A \leftarrow A', B \leftarrow B' \)
for \( i = m+1 \) to \( n \)
if \( w(A) \leq w(B) \)
\( A = A \cup \{i\} \)
else \( B = B \cup \{i\} \)

Approx - Partition is PTAS.

WLOG, assume \( w(A) \geq w(B) \)
approximation ratio \( = \frac{w(A)}{L} \)

\( A \)
\( B \)
\( k \) is the LAST item added to \( A \).
Could have been added in first or second phase.
\[ \frac{1}{2} \leq \frac{s_k}{k} \leq \frac{2L + s_k}{2L} \leq 1 + \frac{s_k}{2L} \]

Since \( s_1, s_2, \ldots, s_m \) are all \( \geq \frac{L}{k} \), we can say that:

\[ \frac{s_k}{k} \leq \frac{L}{k} \]

We know \( k \) is added to \( A \) for the \( m \) items.

This is why \( k \) was added to \( A \).

We have increased after this addition to \( A \).

This means \( A = A' \).

We have an optimal partition.
Approx Vertex Cover - Natural

\[ C \leftarrow \emptyset \]
\[ E' \leftarrow E \]
while \( E' \neq \emptyset \)
    pick \( v \) with maximum degree
    \[ C = C \cup \{v\} \]
    Remove \( v \) and all incident edges from \( E' \)
return \( C \)

A BAD EXAMPLE

\[ k! \text{ vertices of degree } k \]

\[ \frac{k!}{k} \text{ vertices of degree } k \]

\[ \frac{k!}{k-1} \text{ vertices of degree } k-1 \]

\[ k! \text{ vertices of degree 1} \]

Algorithm may end up picking all the bottom vertices
\[ \text{Sol} = k! \left( \frac{1}{k} + \frac{1}{k-1} + \cdots \right) \leq k! \log k. \]
\[ \log k \text{ worse} \]
Approx-Vertex-Cover-Natural is \( \log(n) \)-Approx

\[ |G| = n(\#\text{edges}) \quad G = G_0 \]

\[ G_0 \to G_1 \to G_2 \ldots \quad G_m \text{ with vertex selection \& edge deletion} \]

\[ m = |\mathcal{C}^*| \quad \# \text{vertices in optimal vertex cover} \]

Picking maximum degree vertex of \( G_{i-1} \)

\[ \Rightarrow \text{call the degree } d_i \]

\[ \Rightarrow \text{delete edges incident on picked vertex to get } G_i \]

\[ |G_m| = |G_0| - \sum_{i=1}^{m} d_i \]

\[ \#\text{edges} \]

Also, \( \sum_{i=1}^{m} d_i > \sum_{i=1}^{m} \frac{|G_{i-1}|}{m} \)

(because given \( |G_{i-1}| \) edges can be covered by \( m \) vertices we know there is a vertex with degree at least \( \frac{|G_{i-1}|}{m} \))

\[ \Rightarrow \sum_{i=1}^{m} \frac{|G_m|}{m} \leq \frac{|G_m|}{m} \]

\[ = |G_m| \]

\[ \Rightarrow |G_0| - |G_m| > |G_m| \Rightarrow \text{smaller than } \sum_{i=1}^{m} d_i \Rightarrow m \cdot \log(n) \text{ vertex cover.} \]