1 The Gap Lemmas

In this section we prove two easy lemmas that show how to go from a reduction that causes a gap (like the one in Theorem ??) to obtain a lower bound on approximation algorithms. This first lemma is for max-problems, the second one is for min-problems. The proofs are similar, hence the proof of the second one is omitted.

Def 1.1 Let $f$ be a max-problem (e.g., CLIQ). Let $a(n)$ and $b(n)$ be functions from $\mathbb{N}$ to $\mathbb{N}$ such that $\frac{b(n)}{a(n)} < 1$. Then $\text{GAP}(f, a, b)$ is the following problem.

Problem 1.2

INSTANCE: $y$ for which you are promised that either $f(y) \geq a(|y|)$ or $f(y) \leq b(|y|)$.

QUESTION: Determine which is the case.

Lemma 1.3 Let $A$ be an NP-complete set. Let $f$ be a max-problem. Let $a(n)$ and $b(n)$ be functions from $\mathbb{N}$ to $\mathbb{N}$ such that (1) $\frac{b(n)}{a(n)} < 1$, and (2) $b$ is computable in poly time in $n$. Assume there exists a polynomial time reduction that maps $x$ to $y$ such that the following occurs:

- If $x \in A$ then $f(y) \geq a(|y|)$.
- If $x \notin A$ then $f(y) \leq b(|y|)$.

Then:

1. $\text{GAP}(f, a, b)$ is NP-hard (this follows from the premise).

2. If there is an approximation algorithm for $f$ that, on input $y$, returns a number $> \frac{b(|y|)}{a(|y|)} f(y)$, then $P = NP$.

Proof: We just prove part 2.

We use the reduction and the approximation algorithm to obtain $A \in P$. Since $A$ is NP-complete we obtain $P = NP$.

Algorithm for $A$

1. Input $x$. 
2. Run the reduction on $x$ to get $y$.

3. Run the approximation algorithm on $y$.

4. (This is a comment and not part of the algorithm.)
   \[
x \in A \rightarrow f(y) \geq a(|y|) \rightarrow \text{approx on } y \text{ returns } > \frac{b(|y|)}{a(|y|)} a(|y|) = b(|y|).
   \]
   \[
x \notin A \rightarrow f(y) \leq b(|y|) \rightarrow \text{approx on } y \text{ returns } \leq b(|y|).
   \]

5. If the approx returns a number $> b(|y|)$ then output YES. Otherwise output NO. (This is the step where we need $b(|y|)$ to be computable in polynomial time in $|y|$.)

We now look at min-problems.

**Def 1.4** Let $f$ be a min-problem (e.g., TSP). Let $a(n)$ and $b(n)$ be functions from $\mathbb{N}$ to $\mathbb{N}$ such that $\frac{b(n)}{a(n)} > 1$. Then $\text{GAP}(f, a, b)$ is the following problem.

**Problem 1.5**

- **INSTANCE:** $y$ for which you are promised that either $f(y) \leq a(|y|)$ or $f(y) \geq b(|y|)$.

- **QUESTION:** Determine which is the case.

**Lemma 1.6** Let $A$ be an NP-complete set. Let $f$ be a min-problem. Let $a(n)$ and $b(n)$ be functions from $\mathbb{N}$ to $\mathbb{N}$ such that (1) $\frac{b(n)}{a(n)} < 1$, and (2) $b$ is computable in poly time in $n$. Assume there exists a polynomial time reduction that maps $x$ to $y$ such that the following occurs:

- If $x \in A$ then $f(y) \leq a(|y|)$.
- If $x \notin A$ then $f(y) \geq b(|y|)$.

Then:

1. $\text{GAP}(f, a, b)$ is NP-hard (this follows from the premise).
2. If there is an approximation algorithm for $f$ that, on input $y$, returns a number $< \frac{b(|y|)}{a(|y|)} f(y)$, then $P = NP$. 
**Proof:** We just prove part 2.

We use the reduction and the approximation algorithm to obtain $A \in P$. Since $A$ is NP-complete we obtain $P = NP$.

**Algorithm for $A$**

1. Input $x$.
2. Run the reduction on $x$ to get $y$.
3. Run the approximation algorithm on $y$.
4. (This is a comment and not part of the algorithm.)
   
   $x \in A \rightarrow f(y) \leq a(|y|) \rightarrow \text{approx on } y \text{ returns } < \frac{b(|y|)}{a(|y|)}a(|y|) = b(|y|)$.
   
   $x \notin A \rightarrow f(y) \geq b(|y|) \rightarrow \text{approx on } y \text{ returns } \geq b(|y|)$.
5. If the approx returns a number $< b(|y|)$ then output YES. Otherwise output NO. (This is the step where we need $b(|y|)$ to be computable in polynomial time in $|y|$.)

\[ \Box \]