Inapproximability

Lukáš Poláček
KTH – Royal Institute of Technology, Stockholm

April 24, 2010

1. Introduction

2. P and NP

3. Approximation algorithms
   - PCP-theorem
   - Unique Games Conjecture

What is efficient computation?

- Running time of an algorithm is a function of the length of the input.
- Efficient computation is given by polynomial time computation.
- Running time $n \log n$ or n^3 on an input of length n is acceptable, but 2^n is too much.

P = NP?

- P – polynomial time.
- NP – non-deterministic polynomial time. Given an input x and proof y, program accepts or rejects x depending on y.
- Question of P = NP is unresolved since 1970's. If you solve it, you will get \$1 000 000.

NP-hard problems

- The intuitive notion of a hard problem is that we can use it to solve other problems.
- An NP-hard problem can be used to solve every problem in NP.
- If $P \neq$ NP, then NP-hard problems are not solvable in polynomial time.

Examples of NP-hard problems

- Given a graph, what is the size of the biggest clique?
- Travelling salesman problem – what is the length of the shortest cycle in a graph that visits all vertices?
- 3-SAT: Is CNF formula $\varphi = (x_{11} \vee x_{12} \vee x_{13}) \wedge (x_{21} \vee x_{22} \vee x_{23}) \wedge \cdots \wedge (x_{n1} \vee x_{n2} \vee x_{n3})$ satisfiable?

Approximation algorithms

- Some problems seem to be hard to solve exactly. What if we allowed some error?
- Travelling salesman in Euclidean space allows 2-approximation in polynomial time and $(1 + 1/c)$-approximation in time $O(n(\log n)^{O(c)})$.
- Unfortunately, some NP-hard problems are even hard to approximate.

Probabilistic checkable proofs

Definition (Probabilistic checkable proof)

$A \in \mathrm{PCP}(\log(n), 1)$: given x as an input and y as a proof, A accepts correct proof for correct x with probability 1 and accepts any y for wrong x with probability at most $c < 1$. Moreover, A uses $O(\log n)$ random bits and queries $O(1)$ bits in the proof.

Theorem (PCP theorem, ALMSS 1992)

$\mathrm{PCP}[\log(n), 1] = \mathrm{NP}$

PCP-theorem implications

- Discovery of PCP theorem lead to many tight hardness results.
- Max-Lin-2 is hard to approximate within $2 - \varepsilon$ for any $\varepsilon > 0$.
- Clique is hard to approximate within $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$.

PCP-theorem usage example

Theorem

There is an polynomial algorithm, that transforms a CNF formula φ *to a graph* G, $|G| = 2^q n^b$ *for some constants* q, b, *such that*

- φ *is satisfiable* $\rightarrow Clique(G) \geq n^b$
- φ *is not satisfiable* $\rightarrow Clique(G) < \frac{1}{2} n^b$

Thus if we can approximate Clique within 2, we can solve SAT which is an NP-hard problem.

Unique Games Conjecture (UGC)

- Made by Subhash Khot in 2001 and not proved yet. Implies a lot of tight approximation results.
- Best known algorithm for MaxCut gives 0.878567-approximation, and *UGC* implies that this is optimal [KKMO].

My research goal

- Graph coloring – given a graph, color vertices using the least number of colors such that every two neighboring vertices don't have the same color.
- Current algorithms can color 3-colorable graph with $O(n^{0.202})$ colors.
- On the hardness side, UGC implies that it is hard to color 3-colorable graph with q colors, where q is any positive constant.
- It is one of the few core problems in computer science where there is a big gap between knon approximation algorithms and hardness results.

Thanks for your attention.