Computational Complexity

Lecture 11: Approximation Algorithms

Ronald de Haan
me@ronalddehaan.eu

University of Amsterdam

March 8, 2021

Recap

- Probabilistic algorithms
- Complexity classes BPP, RP, coRP, ZPP

What will we do today?

- Approximation algorithms
- Limits of approximation algorithms

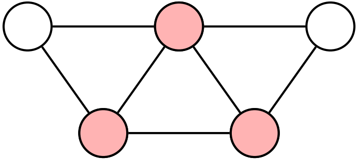
Approximation algorithms

The main idea

- Many NP-complete problems are decision problems asking for an exact/optimal solutions
- *Idea behind approximation:*
  perhaps less than optimal solutions are enough, and easier to compute

Example: Vertex Cover

- Let $G = (V, E)$ be an undirected graph. A subset $C \subseteq V$ is a *vertex cover* of G if each edge in E has at least one endpoint in C.
- Decision problem dec-VC:
  given G and $k \in \mathbb{N}$, does G have a vertex cover of size k?
- We can find the size k_{min} of the smallest vertex cover—and a smallest vertex cover—by calling an algorithm for dec-VC a linear number of times.



Example: Vertex Cover

- Let $G = (V, E)$ be an undirected graph. A subset $C \subseteq V$ is a *vertex cover* of G if each edge in E has at least one endpoint in C.
- Decision problem dec-VC:
  given G and $k \in \mathbb{N}$, does G have a vertex cover of size k?
- We can find the size k_{min} of the smallest vertex cover—and a smallest vertex cover—by calling an algorithm for dec-VC a linear number of times.
- For approximation algorithms, we consider the following problem (say, opt-VC):

  Input: an undirected graph $G = (V, E)$

  Output: a vertex cover $C \subseteq V$ of G

  where we measure the quality of vertex covers C by their size
  (the closer to k_{min}, the better)

Approximation algorithm for Vertex Cover

Definition (Approximation algorithms for VC)

Let $\rho < 1$. A *ρ-approximation algorithm for vertex cover* is an algorithm that, when given a graph $G = (V, E)$ as input, outputs a vertex cover C of G of size at most $1/\rho$ of the minimum size of any vertex cover of G.

- (Sometimes these are called $1/\rho$-approximation algorithms.)

Approximation algorithm for Vertex Cover

- For example, a polynomial-time $^1/_2$-approximation algorithm for vertex cover:

$C := \emptyset$; $G' := G$;
while *G' has edges* **do**
  take some (arbitrary) edge $e = \{v_1, v_2\}$ of G';
  add v_1, v_2 to C and remove all edges containing v_1 or v_2 from G';
end
return C;

- Every edge in G has an endpoint in C, so C is a vertex cover
- The edges $e_1, \ldots, e_m$ used to construct C are pairwise disjoint, and $|C| = 2m$
- Every vertex cover of G must hit each of $e_1, \ldots, e_m$, so must have size $\geq m$

Limits of approximation algorithms

- For vertex cover, we have a polynomial-time $1/2$-approximation algorithm. Can we get a polynomial-time $2/3$-approximation algorithm, or even one for each $\rho < 1$?
- The Cook-Levin Theorem turns out to be not strong enough to rule this out.

Definition (val(φ))

Let φ be a propositional formula in CNF. Then val(φ) is the maximum ratio of clauses of φ that can be satisfied simultaneously by any truth assignment.

Thus, if φ is satisfiable, then val(φ) $= 1$, and if φ is not satisfiable, then val(φ) < 1.

Definition (Approximation algorithms for MAX3SAT)

Let $\rho < 1$. A *ρ-approximation algorithm for* MAX3SAT is an algorithm that, when given a 3CNF formula φ as input, outputs a truth assignment α that satisfies at least a $\rho \cdot \text{val}(\varphi)$ fraction of clauses of φ.

Limits of approximation algorithms

- To rule out ρ-approximation algorithms, we would need something like:
  - If $\varphi \in 3\mathrm{SAT}$, then $\mathrm{val}(\varphi) = 1$
  - If $\varphi \notin 3\mathrm{SAT}$, then $\mathrm{val}(\varphi) < \rho$
- What the Cook-Levin Theorem gives us is a reduction R with:
  - If $x \in L$, then $\mathrm{val}(R(x)) = 1$
  - If $x \notin L$, then $1 - 1/|x| \leq \mathrm{val}(R(x)) < 1$ – you can satisfy all clauses except for one
- So we cannot take any fixed ρ and rule out ρ-approximation algorithms

Definition (PCP verifier)

Let $L \subseteq \{0,1\}^*$ and let $q, r : \mathbb{N} \to \mathbb{N}$ be functions. We say that L has an $(r(n), q(n))$-*PCP verifier* if there is a polynomial-time probabilistic algorithm V with:

- *(Efficiency)* When given as input $x \in \{0,1\}^n$ and when given random access to a string $\pi \in \{0,1\}^*$ of length at most $q(n)2^{r(n)}$ (the *proof*), V uses at most $r(n)$ random coin flips and makes at most $q(n)$ nonadaptive queries to locations of π.
  - Random access: V can query an oracle that gives the i-th bit of π.
  - Nonadaptive queries: the queries do not depend on the answers for previous queries.
- V always outputs either 0 or 1.
- *(Completeness)* If $x \in L$, then there exists a proof $\pi \in \{0,1\}^*$ of length at most $q(n)2^{r(n)}$ such that $\mathbb{P}[\ V^{\pi}(x) = 1\ ] = 1$.
- *(Soundness)* If $x \notin L$, then for every proof $\pi \in \{0,1\}^*$ of length at most $q(n)2^{r(n)}$, it holds that $\mathbb{P}[\ V^{\pi}(x) = 1\ ] \leq 1/2$.

The PCP Theorem (ct'd)

Definition (PCP($r(n), q(n)$))

Let $q, r : \mathbb{N} \to \mathbb{N}$ be functions. The class PCP($r(n), q(n)$) consists of all decision problems $L \subseteq \{0,1\}^*$ for which there exist constants $c, d > 0$ such that L has a $(c \cdot r(n), d \cdot q(n))$-PCP verifier.

Theorem (PCP)

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

- $q(n) = O(1)$, $r(n) = O(\log n)$, so the length $q(n)2^{r(n)}$ of proofs is polynomial
- A constant number $q(n) = O(1)$ of random queries to the proof

The PCP Theorem and approximation algorithms

- The PCP Theorem is equivalent to the following statement:

Theorem (PCP; the approximation view)

There exists some $\rho < 1$ such that for all $L \in \mathrm{NP}$ there is a polynomial-time reduction R from L to 3SAT where for all $x \in \{0,1\}^*$:

- *if $x \in L$ then $val(R(x)) = 1$;*
- *if $x \notin L$ then $val(R(x)) < \rho$.*

- For example: there exists some $\rho < 1$ such that if there exists a polynomial-time ρ-approximation algorithm for MAX3SAT, then $\mathrm{P} = \mathrm{NP}$.

Ruling out polynomial-time ρ-approximation for MAX3SAT for some ρ

- **Statement:** there exists some $\rho < 1$ such that if there exists a polynomial-time ρ-approximation algorithm for MAX3SAT, then $\mathsf{P} = \mathsf{NP}$.
  - Let $L = 3\mathsf{SAT}$. Then there exists some $\rho < 1$ such that there is a polynomial-time reduction R from 3SAT to 3SAT where, for all $x \in \{0,1\}^*$:
    - if $\varphi \in 3\mathsf{SAT}$ then $\mathsf{val}(R(\varphi)) = 1$;
    - if $\varphi \notin 3\mathsf{SAT}$ then $\mathsf{val}(R(\varphi)) < \rho$.
  - Suppose that there exists a polynomial-time ρ-approx. algorithm A for MAX3SAT.
  - We can then solve 3SAT in polynomial time as follows:
    - Take an arbitrary input φ for 3SAT.
    - Produce $\psi = R(\varphi)$ in polynomial time
    - Run A on ψ and count the fraction δ of clauses that are satisfied
    - If $\delta \geq \rho$, then $\varphi \in 3\mathsf{SAT}$; if $\delta < \rho$, then $\varphi \notin 3\mathsf{SAT}$.

Recap

- Approximation algorithms
- Limits of approximation algorithms
- PCP Theorem

Next time

- Subexponential-time algorithms
- The Exponential Time Hypothesis (ETH)

Bonus: polynomial-time $^1/_2$-approximation for MAX3SAT