Carlos Mc Gregor c.mcgregormuro@mail.utoronto.ca

CSC373S: Algorithm Design, Analysis & Complexity

Lecture 33

Wednesday 29, 2017

based on notes by Denis Pankratov

Approximation Algorithms (*cont.*)

Max 3-SAT

Input: φ 3-CNF
the same variable does not appear twice within the same clause.
Output: $max_{x \in \{T,F\}^n} |\varphi(x)|$, where $\varphi(x)$ is the number of clauses satisfied by x
Remember: Input is deterministic; algorithm is random.

For a *randomized algorithm*, approx. ratio is with respect to expected value of the solution of the algorithm compared to opt.

Randomized Algorithm:

```
def MAX-3SAT(phi):
  let n be # of vars in phi

  for i = 1 to n:
    set x_i = T if with probability 1/2 else F

  return x
```

Claim: MAX-3SAT satisfied $7/8$ of the total # of clauses of φ in expectation.
Proof: Let Y be the random var. that is equal to the # of clauses satisfied by the random assignment.

Let $C_1, .., C_m$ be the clauses in φ.
Let

$$Y_i = \begin{cases} 1, & \text{if clause } C_i \text{ is satisfied} \\ 0, & \text{o/w} \end{cases}$$

$Y = \sum_{i=1}^{m} Y_i$

$$\begin{aligned}\mathbb{E}(Y_i) &= 1 \cdot P(Y_i = 1) + O \cdot P(Y_i = 0)\\ &= P(Y_i = 1)\\ &= 7/8\end{aligned}$$

$$\mathbb{E}(Y) = \mathbb{E}(\sum_{i=1}^{m} Y_i)^* = \sum_{i=1}^{m} \mathbb{E}(Y_i) = \sum_{i=1}^{m} 7/8 = (7/8)m \boxtimes$$

* : linearity of expectation, $x_1 \vee \neg x_2 \vee x_3$

Corollary: Max-3SAT algorithm is $8/7$ approx. algorithm (*randomized*).
Derandomization via a method of conditional expectations.

Let the random variable X_i be:

$$X_i = \begin{cases} T, & \text{with probability } 1/2 \\ F, & \text{o/w} \end{cases}$$

$$\begin{aligned}(7/8)m \leq \mathbb{E}(Y) &= \mathbb{E}(Y|X_1 = T)p(x_1 = T) + \mathbb{E}(Y|X_1 = F)p(x_1 = F)\\ &= (1/2)\mathbb{E}(Y|X_1 = T) + (1/2)\mathbb{E}(Y|X_1 = F)\end{aligned}$$

$\Rightarrow$ either $\mathbb{E}(Y|X_1 = T \geq (7/8)m$ or $\mathbb{E}(Y|X_1 = F) \geq (7/8)m$

Observe that we can compute $\mathbb{E}(Y|X_1 = T), \mathbb{E}(Y|X_1 = F)$ efficiently & w/o randomness.

Set X_1 to be T if $\mathbb{E}(Y|X_1 = T > \mathbb{E}(Y|X_1 = F)$, F otherwise.
continue...
Exercise: fill in the details

Weighted Vertex Cover

Input: $G = (V, E)$ undirected graph
$W : V \to \mathbb{Z}_{>0}$-weights
Output: $S \subseteq V$ so that S-vertex covers and $w(S)$ is as small as possible.

Approach: integer program, linear program, etc.

Integer Program for Weighted Vertex Cover (WVC)

Introduce vars $X(v)$ for $v \in V$, min $\sum_{v \in V} w(v) \cdot x(v)$
$\forall \{u, v\} \in E \quad x(u) + x(v) \geq 1$
$\forall v \in E \quad x(v) \in \{0, 1\} \leftarrow$ integrality constraint problem

<u>LP Relaxation:</u>
$\min \sum_{v\in V} w(v)\cdot x(v) \quad (*)$
$\forall\{u,v\}\in E \quad x(u)+x(v)\geq 1$
$\forall v\in E \quad 0\leq x(v)\leq 1 \leftarrow$ no problem!

<u>Algorithm:</u>

```
def WVC-Approx(G=(V,E), w):
  S = []
  solve (*) (from LP Relaxation), let x_hat be the solution

  # Rounding
  for v in V:
    if x_bar(v) >= 1/2:
      S.insert(v)
  return S
```

1. Clearly runs in polytime
2. It always returns a vertex cover

$$\forall\{u,v\}\in E, \bar{x}(u)+\bar{x}(v)\geq 1$$

   $\Rightarrow$ either $\bar{x}(u)\geq 1/2$ or $\bar{x}(v)\geq 1/2$
   $\Rightarrow$ either $u\in S$ or $v\in S$

3. WVC-Approx is a 2-approximation.
   <u>Proof:</u> Let OPT be the minimum weight of a vertex cover.
   Let z be the value of LP on $\hat{x}$.
   $OPT\geq z$ (since z is obtained from the relaxation)

$$\begin{aligned} z &= \sum_{v\in V} w(v)\bar{x}(v) \\ &\geq \sum_{v:\bar{x}(v)\geq 1/2} w(v)\frac{1}{2} \\ &= \frac{1}{2}\sum_{v\in S} w(v) = \frac{1}{2}w(S) \\ &\Rightarrow w(S)\leq 2z\leq 2OPT \quad ☑ \end{aligned}$$

FPTAS (Fully Polynomial Time Approximation Scheme)

$\forall\epsilon>0$ there exists an $(1+\epsilon)$-approx algo that runs in time polynomial in the inut size & $1/\epsilon$