Lecture 3

Saturday, March 25, 2017 3:55 PM

Problems

① Max-Cut

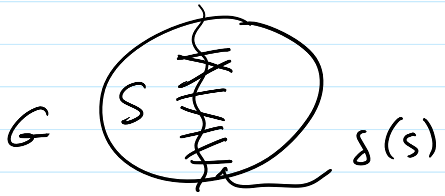
Input: $G=(V,E)$, wts $w(e)$ on edges

Output: $S \subseteq V$

Objective: maximize $w(\delta(S))$

$\uparrow$

$\left\{ e=(u,v) \,\middle|\, \begin{matrix} u \in S, v \notin S \\ \text{or} \\ u \notin S, v \in S \end{matrix} \right\}$



② Max-k-coverage

Max-Cut

Algorithm:

- Start with an arbitrary set $\overset{S}{\overset{\shortparallel}{S_0}} \subseteq V$
- While $\exists\, v \in S$ st $w(\delta(S-v)) > w(\delta(S))$
  or
  $\exists\, v \notin S$ st $w(\delta(S+v)) > w(\delta(S))$
  perform the "local move"

Analysis

At the end of the while loop, we will end with a set S s.t

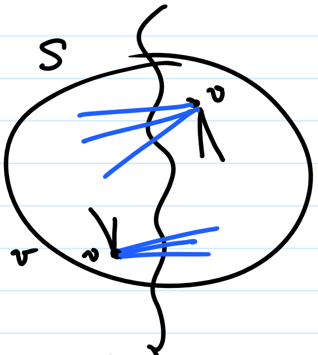
$$\begin{cases} \forall v \in S, & w(\delta(S-v)) \leq w(\delta(S)) \\ \forall v \notin S, & w(\delta(S+v)) \leq w(\delta(S)) \end{cases}$$

$(\forall v \notin S,\ w(\partial(S+v)) \leq w(\partial(S)))$

$\Downarrow$

$\forall v,$

$\partial(v)$: edges inc. on v



$w(\partial(v)\cap\partial(S))$
$\geq w(\partial(v)\setminus\partial(S))$.

Add this for all v.

$$\sum_v w(\partial(v)\cap\partial(S)) \geq \sum_v w(\partial(v)\setminus\partial(S))$$

$$\parallel$$

$$\cancel{2}\cdot w(\partial(S)) \geq \cancel{2}\, w(E\setminus\partial(S))$$

$\because$ each edge $(u,v)\in\partial(S)$ is counted once in $\partial(u)$ & once in $\partial(v)$

$\because$ all edges not in $\partial(S)$ are counted exactly twice.

$$\Rightarrow \quad w(\partial(S)) \geq w(E) - w(\partial(S))$$

$$\Rightarrow \quad w(\partial(S)) \geq \frac{1}{2}\cdot w(E) \geq \frac{1}{2}\cdot OPT$$

Thm: Upon termination, the algorithm returns a 2-appx. max-cut.

Running time? Suppose all the weights were integers. Then everytime the MAX-CUT grows by at least 1. So if all the weights are $\leq$ poly(n), then the algo terminates in polynomial time.

But the weights can be as large as

$2^{\text{poly}(n)}$ … representation is still poly(n) bits.

"**Standard Fix**":

- Take a hit in the factor by ε.

→ $$w(\partial(S-v)) \geq \left(1+\frac{\varepsilon}{n}\right) w(\partial(S)) \quad ; v \in S$$

or

$$w(\partial(S+v)) \geq \left(1+\frac{\varepsilon}{n}\right) w(\partial(S)) \quad ; v \notin S$$

Exercise :- Upon termination.

$$w(\partial(S)) \geq \left(\frac{1}{2} - O(\varepsilon)\right) \cdot OPT$$

Running Time : $\leq \log_{1+\varepsilon/n} nW$

where $W := \max_e w(e)$

Max-Coverage.

Input :- Sets $S_1, S_2, \ldots, S_m \subseteq \mathcal{U}$.

Output : k of these sets.

Objective :- Maximize cardinality of union.

Local Search Algorithm:

→ Start with any collection of k sets

→ If there exists a "swap" which increases the union, do it.

More precisely, Let $|A| = k$ be the current indices. If $\exists\ a \in A,\ b \notin A$ s.t

$$\left|\bigcup_{i\in A} S_i\right| < \left|\bigcup_{i\in A\setminus a} S_i \cup S_b\right|$$

Then $A = A - a + b$

→ Since no weights are involved, this terminates in n-steps. to a local optimum set A satisfying :

$$(*)\quad \left|\bigcup_{i\in A} S_i\right| \geq \left|\bigcup_{i\in A\setminus a} S_i \cup S_b\right|$$

for all $\begin{array}{l} a\in A \\ b\notin A \end{array}$

Analysis

- Let O be the optimal coverage solⁿ.

$$\left.\begin{array}{l} -\ O = \{o_1, o_2, \dots, o_k\} \\ -\ A = \{a_1, a_2, \dots, a_k\} \end{array}\right\}\ \text{arbitrarily re-named.}$$

- Let $ALG = \bigcup_{i=1}^{k} S_{a_i}$, $OPT = \bigcup_{i=1}^{k} S_{o_i}$

Let's define for $j = 1 \dots k$

$$ALG_{-j} := \bigcup_{i\neq j} S_{a_i}$$, ie, all the elts covered by sets other than

- $S_{a_j} \setminus ALG_{-j}$: the elts solely covered by S_{a_j}

- Local opt $\Rightarrow$ (convince yourself)

$$|S_{a_j} \setminus ALG_{-j}| \geq |S_{o_j} \setminus ALG_{-j}|$$

$\forall j = 1 \ldots k$

$$\rightarrow \quad \underbrace{\sum_{j=1}^{k} |S_{a_j} \setminus ALG_{-j}|}_{\text{elements in ALG covered by exactly one set.}} \leq |ALG| \quad ①$$

$$\rightarrow \quad \sum_{j=1}^{k} |S_{o_j} \setminus ALG_{-j}|$$

$$\geq \sum_{j=1}^{k} |S_{o_j} \setminus ALG|$$

$\because ALG_{-j} \subseteq ALG$

$$\geq \left| \bigcup_{j=1}^{k} S_{o_j} \setminus ALG \right|$$

also called the "UNION BND" : $|A|+|B| \geq |A \cup B|$

$$= |OPT \setminus ALG|$$

$$\geq |OPT| - |ALG| \quad ②$$

Putting together ① & ②

$$|ALG| \geq \frac{1}{2} \cdot |OPT|$$

Thm: The local search algorithm is a $\frac{1}{2}$-appx for Max-k-coverage.

Last class we saw a $(1-\frac{1}{e})$-factor algorithm for max-k-coverage. Why is this $1/2$-appx so interesting?

Because our analysis also holds for more sp. cases.

Suppose the sets $S_1, S_2, \ldots, S_m$ were "colored" in k-diff. colors. That is, there is a color fn. $col: \{1, \ldots m\} \to \{1, \ldots, k\}$ spec. the color of each set

"Colorful Max k-coverage":

Pick one set from each color to maximize the union.

The analysis of GREEDY breaks down.

Convince Yourself. Infact show that GREEDY can't get $1-\frac{1}{e}$ ....

... what does it get?

How about the local search algorithm?
Now when we perform swaps we have to be careful to respect color classes.

CRUX: Analysis goes through "fly-to-fly" since the $A = \{a_1, \ldots, a_k\}$ & $O = \{o_1, \ldots, o_k\}$ can be aligned s.t. a_i & o_i are in the same col--class.

i.e. $A - a_i + o_i$ is valid.

More generally:

Defn: A set system $(V, \mathcal{I})$ is a matroid with $\mathcal{I}$ being the independent sets, if

(mono) ① $\forall A \in \mathcal{I},\ B \subseteq A \Rightarrow B \in \mathcal{I}$

(exchange) ② $A \in \mathcal{I}, B \in \mathcal{I}$ & $|A| < |B|$, then $\exists\, i \in B \setminus A$ s.t. $A + i \in \mathcal{I}$

Many excellent properties, Many examples

① $V \equiv$ cols of a matrix
$\mathcal{F} \equiv$ sets of lin. ind. col's.

② $V \equiv$ Edges of a graph
$\mathcal{F} \equiv$ sets not containing cycles.

③ $V \equiv \{1, 2, \ldots, n\}$
$\mathcal{F} \equiv$ Sets of card $\leq k$

④ $V \equiv \{1, 2, \ldots, n\}$
$c : \{1, \cdots, n\} \to \{1, 2, \ldots, k\}$

$\mathcal{F} \equiv$ sets containing ≤ 1 elt from each class.

$\vdots$

In general one can look at the

<u>Matroid Max-k-coverage problem</u>

<u>Input</u> • Sets $S_1, \cdots, S_m$

• matroid $\mathcal{M} := ([m], \mathcal{F})$

- matroids $\mathcal{M} := ([m], \mathcal{F})$

Output : $A \in \mathcal{F}$

Objective : $\max \left| \bigcup_{i \in A} S_i \right|$

Local search algorithm is similar except when we swap out we must be careful to be in $\mathcal{F}$.

The analysis is once again goes "f-to-f" because of the following remarkable theorem.

Exchange Theorem for Matroids

Given any two maximal independent sets $B_1, B_2 \in \mathcal{F}$, there is a bijection

$$\phi : B_1 \to B_2 \quad \text{s.t}$$

$$\forall\, i \in B_1, \quad B_1 - i + \phi(i) \in \mathcal{F}$$

Once we know this, again A and O can be "lined up" using this bijection.

$$A = \{a_1, \ldots, a_k\}$$

$$O = \{\phi(a_1), \phi(a_2), \dots, \phi(a_k)\}$$

$$\boxed{\text{Thm: Local Search is a } \tfrac{1}{2}\text{-appx for Matroid Max-}k\text{-coverage.}}$$

---

k-Median:

Input: • Metric Space (X, d)

• $X = \underset{\substack{\nearrow \\ \text{facilities}}}{F} \cup \underset{\substack{\nwarrow \\ \text{clients}}}{C}$

Output: • $\underset{\substack{\uparrow \\ \text{facilities opened by algorithm.}}}{A} \subseteq F$, $|A| = k$

Given A, for every $j \in C$ let $A(j) \in A$ be the nearest facility to j in A.

Objective: $\underset{F \supseteq A : |A| = k}{\text{Minimize}} \quad \sum_{j \in C} d(j, A(j)) =: \text{cost}(A)$

ALGORITHM

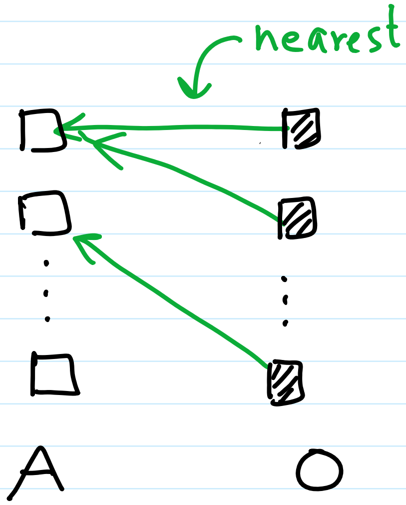
```
• Start with an arbitrary $A_0 \subseteq F$, $|A_0| = k$
• While $\exists\, i \in A$, $i' \notin A$ s.t.
      $\text{cost}(A - i + i') < \text{cost}(A)$
    – $A = A - i + i'$
```

Local opt : A s.t. $\text{cost}(A - i + i') \geq \text{cost}(A)$
$\forall i \in A,\ i' \notin A.$

Analysis : Let $O \subseteq F$ be the optimum soln.

$$OPT = \sum_{j \in C} d(j, O(j))$$

Pairing-Up :



Notation : • $\forall i \in A$, let $\Gamma_i \subseteq C$ be the set of clients that are assigned to i in the opt soln.

- • $\forall i^* \in O$, Γ_{i^*} is analog. set.

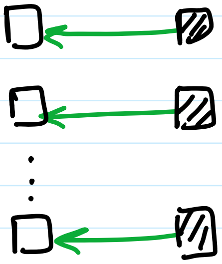
- $\forall j \in C$, let $c(j)$ = cost it pays in A
$= d(j, A(j))$

$\&\quad c^*(j) = d(j, O(j))$

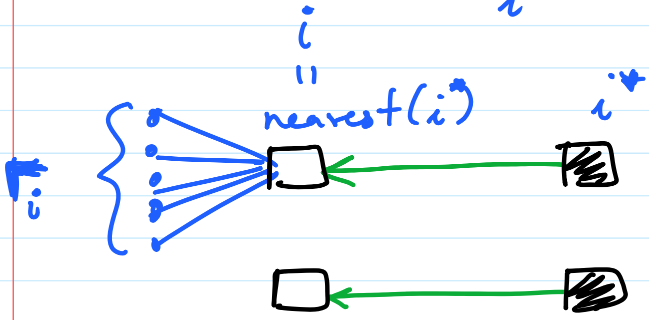
- $\therefore OPT = \sum_{j \in C} c^*(j)$; $ALG = \sum_{j \in C} c(j)$

- ∴ $OPT = \sum_{j \in C} c^*(j)$; $ALG = \sum_{j \in C} c(j)$
- $\underline{\forall i \in O}$: nearest(i) is the facility in A which is nearest to i.

  The "green edges" above show the nearest map.
- For the time being, suppose "nearest" was $1 \Leftrightarrow 1$. NOT WITHOUT LOSS OF GEN.



- Write the Local-Opt conditions for $\text{SWAP}(\underbrace{\text{nearest}(i^*)}_{i}, i^*)$



Let's find an assignment of clients to fac in $(A - i + i^*)$. We know that any such assignment has cost $\geq$ cost(A).

— Interesting set : $(\Gamma_i \cup \Gamma_{i^*})$

- assign every client in Γ_{i^*} to i^*

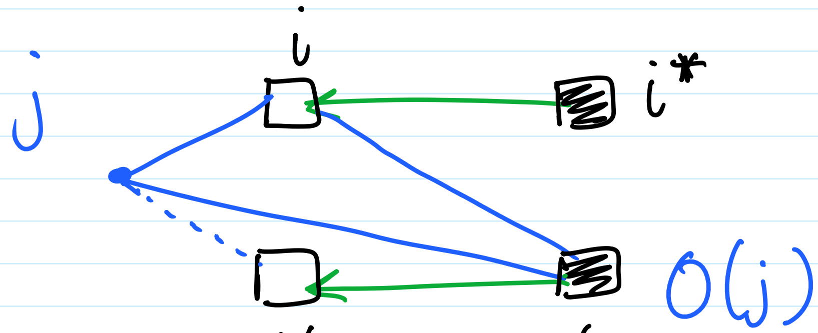
  Why? ∵ we want after all to comp. with opt.

- For every client $j \in \Gamma_i$, we need to find a facility.

Let's see where j went to in OPT.

j went to $O(j)$

- If $O(j) = i^*$, then HAPPY
  (send $j \rightsquigarrow i^*$)
- If $O(j) \neq i^*$
  Send $j \rightsquigarrow \text{nearest}(O(j))$



$i' = \text{nearest}(O(j))$

Since we assumed 1-1 mapping,
nearest $(O(j)) \neq i$ and is ∴ available.

How do we bound $d(j, i')$??

$$d(j,i') \le d(j,O(j)) + d(i',O(j))$$

∵ Δ-ineq

$$\le d(j,O(j)) + d(i,O(j))$$

∵ i' was nearest to $O(j)$

$$\le d(j,o(j)) + d(i,j) + d(j,o(j))$$

again Δ-ineq.

$$= 2d(j,o(j)) + d(j,i)$$

$$= 2c^*(j) + c(j)$$

Ok, so what did we show?

If we close i & open i^*, then there is a way to assign the clients in $(\Gamma_i \cup \Gamma_{i^*})$ which incurs a <u>cost</u> of

$$\le \sum_{j\in\Gamma_{i^*}} c^*(j) + \sum_{j\in\Gamma_i\setminus\Gamma_{i^*}} \left(2c^*(j) + c(j)\right)$$

Since A was locally opt., this <u>cost</u> must be at least what these clients pay in A

$$\therefore \sum_{j\in\Gamma_i\cup\Gamma_{i^*}} c(j) \le \sum_{j\in\Gamma_{i^*}} c^*(j) + \sum_{j\in\Gamma_i\setminus\Gamma_{i^*}} \left(2c^*(j) + \cancel{c(j)}\right)$$

$$\Vert$$

$$\sum_{j\in\Gamma_{i^*}} c(j)$$

$$+$$

$j \in \Gamma_{i^*}$

$$+ \cancel{\sum_{j \in \Gamma_i \setminus \Gamma_{i^*}} c(j)}$$

$$\Rightarrow \sum_{j \in \Gamma_{i^*}} c(j) \leq \sum_{j \in \Gamma_{i^*}} c^*(j) + 2 \sum_{j \in \Gamma_i \setminus \Gamma_{i^*}} c^*(j)$$

Summing up for <u>all</u> $i^* \in O$

$$ALG \leq OPT + 2 \sum_{i^*} \sum_{j \in \Gamma_{\underset{\underset{\text{nearest}(i^*)}{\nearrow}}{i}} \setminus \Gamma_{i^*}} c^*(j)$$

$$\leq OPT + 2 \sum_{j \in C} c^*(j)$$

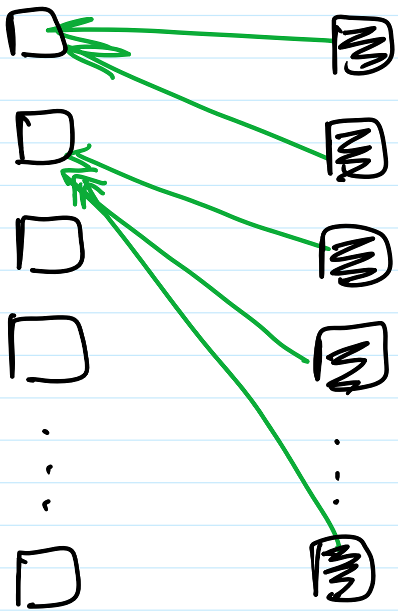
Since nearest is $1 \leftrightarrow \underline{1}$
each client j is
at most counted <u>once</u>.

$$= 3 \cdot OPT$$

∴ If the mapping "nearest" was

1 – 1 (and there is no reason why it should be), then the Local opt soln is within 3-times the global opt soln.

What if "nearest" is not $1 \leftrightarrow 1$?



$\underline{Q^n}$: ① Which pairs should we swap?

② Where all did we use 1-1 in the above analysis?

"if (i^*, i) are swapped, then we needed that nearest(i'') for every other $i'' \in O \setminus i^*$ <u>must be</u> present in $A \setminus i$."