Approximation Algorithms

Maurice Queyranne

Lecture 2

August 4, 2009

Summary of Lecture 1

- notions of approximation algorithm (for NP-hard optimization problems)
  dual problem and weak duality
- Vertex Cover: 2 approximation algorithm
  - Greedy algorithm (maximum degree) ... ?
  - Maximal Matching Based VC alg: approx. ratio 2
- Metric Steiner problem:
  - Shortest Spanning Tree based on shortest distances: approx. ratio 2
- Traveling Salesman Problem (TSP)
  with arbitrary (nonnegative) lengths, there cannot be any
  approximation bound (unless P = NP)

Metric TSP:

given the complete graph K_n on n nodes, and edge lengths $\ell_{ij} \geq 0$ satisfying the Δ-inequality

$$\ell_{ij} + \ell_{jk} \geq \ell_{ik} \quad \text{for all } i,j,k \text{ in } V(K_n)$$

Double-Tree TSP algorithm:

- find a Shortest Spanning Tree T^* in G
- duplicate each edge in T^*: yields T^{**}
- find a Eulerian cycle C in T^{**}
- start at any node v, follow C taking shortcuts to avoid visiting a vertex twice, until all vertices are visited
- return to v

<u>Theorem</u>: The Double Tree TSP algorithm is a 2-approximation

proof: a tour contains a Hamiltonian path P, which is a tree, so

$$OPT \geq \text{length}(P) \geq \text{length}(T^*)$$

let C^{DT} be the cycle produced by the Double Tree TSP alg.

$$\text{length}(C^{DT}) \underset{\Delta\text{-ineq.}}{\leq} \text{length}(C) = \text{length}(T^{**}) = 2\,\text{length}(T^*) \leq 2\,OPT$$

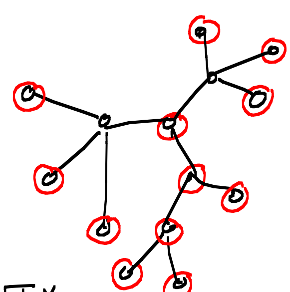
QED

Instead of duplicating every edge in T^*, add a set M of edges that forms a <u>perfect matching</u> of the odd-degree vertices (in T^*)

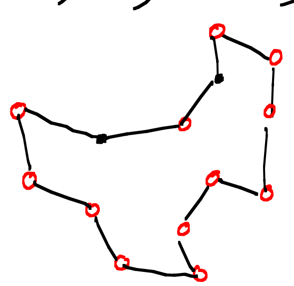
i.e. look at the complete subgraph G' induced by these odd-degree vertices, with the original edge lengths,
and find a perfect matching M^* in G' with least total length.
- can be solved in polytime $O(|V(G')|^3)$

<u>Lemma</u>: the total length of such an optim perfect matching M^* satisfies: $\text{length}(M^*) \leq \frac{1}{2}\,OPT$

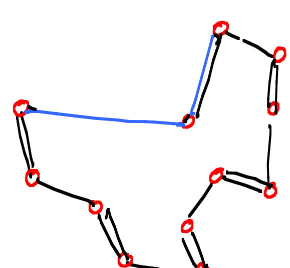
proof: Take an optim tour (Hamiltonian cycle with shortest length in the original K_n), shrink this tour to the odd-degree vertices (in T^*) by taking shortcuts



T^*



optim tour



shrunk tour

$= M'$

$- M''$

Recall that there is an even number of odd-degree vertices (because the sum of all degrees is twice the total number of edges, hence even)

hence we can decompose the shrunk tour into two perfect matchings M' and M'' of these odd-degree vertices

$$OPT \underset{\Delta\text{-ineq.}}{\geq} \text{length(shrunk tour)} = \text{length}(M') + \text{length}(M'') \geq 2\,\text{length}(M^*)$$

QED

Christofides TSP algorithm for Metric TSP

- find a shortest spanning tree T^*
- find a minimum length perfect matching M^* of the odd-degree vertices (in T^*)
- find a Eulerian cycle C in the (Eulerian) graph $(V(K_n), T^* \cup M^*)$
- start at any node v, follow C taking shortcuts to avoid visiting a vertex twice, until all vertices are visited

<u>Theorem</u>: Christofides TSP algorithm is a $3/2$-approximation for the metric TSP

Review of Linear Programming Duality

used to find dual bounds for a Linear Programming problem

example: $$\mathrm{OPT} = \left[\begin{array}{lll} \min & 130x_1 + 370x_2 + 500x_3 & \\ \text{s.t.} & 0.1x_1 + 0.4x_2 \geq 4 & (1) \\ & 0.2x_1 + 0.3x_2 + x_3 \geq 5 & (2) \\ & x \geq 0 & \end{array}\right.$$

How to find dual (lower) bounds on OPT?

- $\mathrm{OPT} \geq \underline{0}$ because $x \geq 0$ and all objective coefficients are ≥ 0
- $500 * (2)$: $130x_1 + 370x_2 + 500x_3 \geq 100x_1 + 150x_2 + 500x_3 \geq \underline{2{,}500}$

  $\left(\begin{array}{l} x \geq 0,\ 130 \geq 100 \\ 370 \geq 150 \end{array}\right)$

- add $300 * (1)$: $$\begin{array}{r} 30x_1 + 120x_2 \geq 1{,}200 \\ \hline 130x_1 + 270x_2 + 500x_3 \geq \underline{3{,}700} \end{array}$$

How to find the best possible dual bound using this approach (i.e., nonnegative combination of the inequality constraints)?

find multipliers y_1 for (1) and y_2 for (2) such that

- $y_1, y_2 \geq 0$ (to preserve the direction of the inequalities)
- the coefficient of each variable x_j in the combination is at most its objective coefficient (because $x_j \geq 0$):

$$\left.\begin{array}{lrcl} x_1: & y_1\,0.1 + y_2\,0.2 & \leq & 130 \\ x_2: & y_1\,0.4 + y_2\,0.3 & \leq & 370 \\ x_3: & y_2 & \leq & 500 \end{array}\right\} (3)$$

so that the resulting lower bound $y_1 4 + y_2 5$ is as large as possible

this is a LP: $\left[\begin{array}{l} \max\ y_1 4 + y_2 5 \\ \text{s.t. } (3) \text{ and } y \geq 0 \end{array}\right.$ called the **dual** of the given LP

Generally, given a LP problem (P) with variables indexed by J and constraints indexed by I we can define its dual, a LP in which

- there is one variable y_i for every constraint ($i \in I$) in (P)
  the sign of y_i depends on the direction of this constraint (so we obtain a dual bound)
- there is one constraint for every variable x_j ($j \in J$) in (P)
  - its direction depends on the sign restriction on x_j
  - its right hand side (RHS) is the objective coefficient of x_j in (P)
- the objective coefficients are the RHS of the constraints in (P)
  the direction of optimization is opposite of that in (P)

Given a LP problem $\min\{cx : Ax \geq b,\ x \geq 0\}$ (P)
its dual is $\max\{yb : yA \leq c,\ y \geq 0\}$ (D)

Lemma (Weak LP Duality)

Let x be any feasible solution to (P)
y — — — — — its dual (D)
Then $cx \geq yb$

Proof: if x, y are feasible then

$$cx - yb = cx - yAx + yAx - yb = \underbrace{(c-yA)}_{\geq 0}\underbrace{x}_{\geq 0} + \underbrace{y}_{\geq 0}\underbrace{(Ax-b)}_{\geq 0} \geq 0$$

QED

Strong LP Duality Theorem:

If a LP problem is feasible and has a bounded objective value (in the direction of optimization) then it has an optimum solution, its dual has an optimum solution, and their objective values are equal

Optimality Property (Complementary Slackness Conditions)

If x and y are feasible solutions to a LP (P) and its dual (D), respectively, then x and y are optimum if and only if

(CS1) for every constraint in (P) either the constraint is tight for x (Left hand side for x equals RHS) or the dual variable is 0 (or both) and

(CS2) for every variable in (P) either the variable is 0 at x or the dual constraint is tight at y (or both)

for (P) and (D) above:

(CS1) $\underbrace{A_{i.}}_{i\text{-th row of } A} x = b_i$ or $y_i = 0$ (or both)

$$y_i(A_{i.}x - b_i) = 0 \quad \forall i \in I$$

(CS2) $x_j = 0$ or $y \underbrace{A_{.j}}_{j\text{th column of } A} = c_j$ (or both)

$$(c_j - yA_{.j})x_j = 0 \quad \forall j \in J$$

proof: if x and y are optimum

$$0 = cx - yb = \sum_{j \in J} \underbrace{(c_j - yA_{.j})}_{\geq 0} \underbrace{x_j}_{\geq 0} + \sum_{i \in I} \underbrace{y_i}_{\geq 0} \underbrace{(A_{i.}x - b_i)}_{\geq 0}$$

iff every term $(c_j - yA_{.j})x_j = 0$ and $y_i(A_{i.}x - b_i) = 0$ QED

Covering problems, and Greedy algorithms

- recall the Vertex Cover problem
  - (Max Degree) Greedy algorithm

    we will see that its approximation ratio is $H(n)$

    where n = number of vertices

    $H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{i=1}^{n} \frac{1}{i}$ is the n-th harmonic number

    recall $\frac{1}{n} + \log(n) \leq H(n) \leq 1 + \ln(n)$

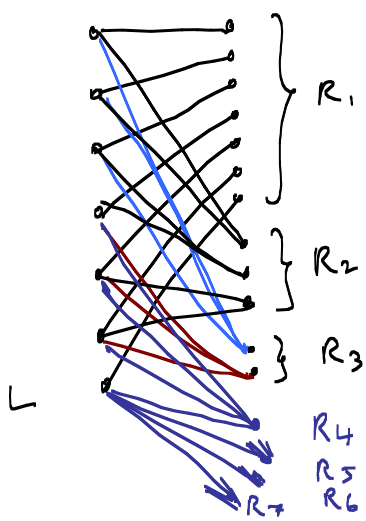
  - asymptotically tight instances

    bipartite graph $G = (L \dot\cup R, E)$

    $|L| = n$, $R = R_1 \dot\cup R_2 \dot\cup \dots \dot\cup R_n$ and each $|R_k| = \left\lfloor \frac{n}{k} \right\rfloor$

    each vertex in R_k is connected to k vertices in L

    and two distinct vertices in the same R_k are connected to different vertices in L



so $|R| = n + \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{n}{3} \right\rfloor + \dots + 1 \approx H(n)$

and every vertex in R_k has degree k

every vertex in L is connected to at most one vertex in each R_k, so degree $\leq n$

Greedy algorithm:

- choose vertex in R_n (max degree)

  so max degree in remaining graph is $n-1$

- choose all vertices in R_{n-1} (one by one), etc.

outputs R with size $|R| \approx n H(n)$

but L is a vertex cover, so $OPT \leq |L| = n$

so $|R| \approx H(n)\, OPT$

Set Cover

given a finite set N (universe, ground set ...)

a collection of subsets $S \subseteq N$ each with cost c_S

find a subcollection $\mathcal{S}$ of these subsets, that covers N

(every element of N is in at least one $S \in \mathcal{S}$, i.e., $\bigcup_{S \in \mathcal{S}} S = N$)

with minimum total cost $c(\mathcal{S}) = \sum_{S \in \mathcal{S}} c_S$

examples:

- Vertex Cover is the special case where each given subset S contains exactly two elements
- in Vehicle Routing, we want to find a collection of routes that, together, visits all customers in set N, at minimum total cost: the (given) sets S are all possible subsets of customers that can be on a single route

  c_S is the minimum cost of a route that visits all customers in S

  Remark: routes may be generated "on the fly", by Column Generation
- other applications in location (N is the set of points to be serviced, each S the sets of points that can be serviced by a facility in a given location); network design, etc..

Integer Programming Formulation of Set Cover

decision variables $x_S = \begin{cases} 1 & \text{if set } S \text{ is selected} \\ 0 & \text{o/w (otherwise)} \end{cases}$

$$\begin{cases} \min \sum_S c_S x_S \\ \text{s.t.} \sum_{S : i \in S} x_S \geq 1 \qquad \forall \text{ element } i \in N \\ \qquad 0 \leq x \leq 1, \; x \text{ integer} \end{cases}$$

more compactly:

(SC) $$\begin{cases} \min & cx \\ \text{s.t} & Ax \geq \underline{1} \\ & 0 \leq x \leq 1 \\ & x \text{ integer} \end{cases}$$

where A is an $m \times n$ matrix with 0-1 entries
$\underline{1}$ is an n-vector of 1's
$n = |N|$
$m =$ number of given subsets

(in the Vertex Cover special case, every column of A has exactly two nonzero entries)

Integer Cover:

given an $m \times n$ matrix A with nonnegative integer entries
an m-vector b — positive — —
an n-vector c of — objective coefficients

find a binary vector $x \in \mathbb{B}^n$ where $\mathbb{B} = \{0,1\}$

to $$\begin{cases} \min & cx \\ \text{s.t.} & Ax \geq b \\ & 0 \leq x \leq 1, \ x \text{ integer} \end{cases}$$

examples:

- Set Cover (where A is a 0-1 matrix, b and c are vectors of 1's)
- Cutting Stock problem:

  given orders b_i units of type i objects
  sheets that be used to cut objects, with unit cost c_j

  select sheets to be c - to produce all required objects, at minimum cost define a pattern P_j as a sheet and a set of objects that be cut from it

let a_{ij} = number of type-i objects in pattern P_j

$$\left[\begin{array}{ll} \min & \sum_j c_j x_j \\ \text{s.t.} & \sum_j a_{ij} x_j \ge b_i \quad \forall \text{objects } i \\ & 0 \le x \le 1 \\ & x \text{ integer} \end{array}\right.$$

(patterns can be generated using Column Generation)

Submodular Set Cover (SSC) problem

a submodular (set) function $f : 2^N \to \mathbb{R}$

where 2^N is the set of all subsets of given finite set N

satisfies the submodular inequality

$$(SI) \quad f(S \cup T) + f(S \cap T) \le f(S) + f(T) \quad \forall S, T \subseteq N$$

Remark: (SI) is equivalent to nonincreasing marginal values

$$f(A \cup \{i\}) - f(A) \ge f(B \cup \{i\}) - f(B) \qquad \forall A \subseteq B \subseteq N \setminus \{i\}$$

(decreasing returns of scale; the elements of N are "substitutes")

- f is supermodular if $-f$ is submodular (i.e., the above inequalities are reversed)

for a supermodular function, the elements of N are "complements"

(SI) is also equivalent to "local submodularity":

$$f(A \cup \{i,j\}) - f(A \cup \{j\}) \le f(A \cup \{i\}) - f(A) \quad \forall A \subseteq N \setminus \{i,j\},\ \forall i \ne j$$

A set function $g: 2^N \to \mathbb{R}$ is **nondecreasing** if
$$g(S) \le g(T) \quad \forall S \subseteq T \subseteq N$$

Examples:

1) given numbers $a_i \in \mathbb{R} \ \ \forall i \in N$

$a(S) = \sum_{i \in S} a_i$ defines a function which is both submodular and supermodular, i.e., a **modular** function (the inequality in SI holds as equality)

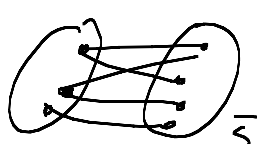
it is nondecreasing iff all $a_i \ge 0$

2) a **cut function**:

given a graph $G = (N, E)$

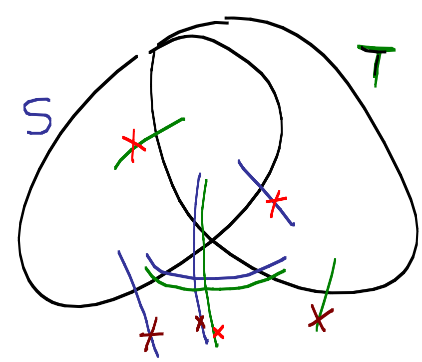
$\forall S \subseteq N$, $f(S)$ = number of edges with exactly one endpoint in S

(Example: if $G = K_n$ then $f(S) = |S| \cdot (n - |S|)$ which is <u>not</u> nondecreasing)

S  $\bar{S} = N \setminus S$

this cut function is submodular

	$S \setminus T$	$S \cap T$	$T \setminus S$	$N \setminus (S \cup T)$
$S \setminus T$		✓	✓✓	✓
$S \cap T$	✓		✓	✓✓
$T \setminus S$	✓✓	✓		✓
$N \setminus (S \cup T)$	✓	✓✓	✓	

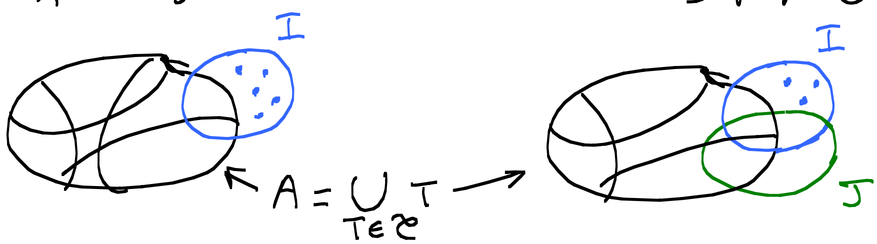


$$f(S) + f(T) = f(S \cup T) + f(S \cap T) + 2\left|\{\{i,j\} \in E : i \in T \setminus S,\ j \in S \setminus T\}\right|$$

3) in Set Cover, given ground set N and collection $\mathcal{S}$ of subsets of N, let $f : 2^{\mathcal{S}} \to \mathbb{R}$ defined by

$$f(\mathcal{T}) = \Big| \bigcup_{T \in \mathcal{T}} T \Big|$$ the number of elements covered by subsets in $\mathcal{T}$

- f satisfies the local submodularity property



$f(\mathcal{T} \cup \{I\}) - f(\mathcal{T}) = |I \setminus A| \qquad f(\mathcal{T} \cup \{i,j\}) - f(\mathcal{T} \cup \{j\}) = |I \setminus (A \cup J)| \leq |I \setminus A|$

4) In the Integer Cover

given a nonnegative integer matrix A, and positive RHS vector b

N is the set of columns of A, $M = \{1, \dots, m\}$ its rows

$$g(S) = \sum_{i=1}^{m} \min \Big\{ \sum_{j \in S} a_{ij} \, , \, b_i \Big\}$$

(Example: the function f of the previous example, Set Cover)

- g is nondecreasing and submodular

5) (OPTIONAL MATERIAL)

The rank function r of a matroid is a nondecreasing and submodular set function (satisfying $r(\emptyset) = 0$ and $r(S \cup \{i\}) - r(S) \in \{0, 1\}$ for all S, i)

Submodular Set Cover (SSC) problem

given a ground set N

cost $c_j > 0$ of elmt $j \in N$

a nondecreasing submodular function $f : 2^N \to \mathbb{R}$ with $f(\emptyset) = 0$

find a subset $S \subseteq N$ such that $f(S) = f(N)$

with least possible cost $c(S) = \sum_{j \in S} c_j$

i.e., $\min_{S \subseteq N} \{ c(S) : f(S) = f(N) \}$

- by Example 4 above, Integer Cover (and therefore its special cases, Set Cover, Vertex Cover) is a special case of SSC with $f = g$ (assuming $\sum_{j \in N} a_{ij} \geq b_i \;\; \forall i$, otherwise the problem is infeasible)

indeed: $g(N) = \sum_{i=1}^{m} b_i$

and $g(S) = g(N)$ iff $\sum_{j \in S} a_{ij} = b_i$ for all $i = 1..m$