CSC 2421 - Approximation Algorithms

Lecture #4: 28 January 2003

Lecturer: A. Borodin

Scribe Notes by: A. King and M. Lou

1 Agenda

- The primal/dual approach
- Facility Location Problem and a primal/dual algorithm
- Claim: All one-pass primal-dual algorithms are priority algorithms

2 Review: Primal/Dual Linear Programs

Our standard linear program is expressed as follows:

Primal:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} c_i x_i \\ \text{subject to} & \sum_{i=1}^{n} a_{ij} x_i \geq b_j \text{ for } 1 \leq j \leq m \\ & x_i \geq 0 \text{ for } 1 \leq i \leq n \\ & x_i \in \{0,1\} \text{ for the integer case.} \end{array}$$

Dual:

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{m} b_j y_j \\ \text{subject to} & \sum_{i=1}^{m} a_{ij} y_j \leq c_i \text{ for } 1 \leq i \leq n \\ & y_j \geq 0 \text{ for } 1 \leq j \leq m \end{array}$$

3 Primal/Dual Slackness Conditions

x and y are optimal primal/dual linear program solutions if and only if:

- $x_i = 0$ or $\sum_{i=1}^{m} a_{ij} y_j = c_i$ (primal slackness)
- $y_j = 0$ or $\sum_{i=1}^{n} a_{ij} x_i = b_j$ (dual slackness)

Our relaxed slackness conditions are, for some $\alpha \geq 1$ and $\beta \geq 1$,

- $x_i = 0$ or $c_i/\alpha \leq \sum_{i=1}^{m} a_{ij} y_j \leq c_i$ (primal slackness)

- $y_j = 0$ or $b_j \leq \sum_{i=1}^{n} a_{ij}x_i \leq \beta b_j$ (dual slackness)

If x and y are primal/dual feasible, x is integral, and relaxed slackness conditions hold, then x is an $\alpha\beta$ approximation:

$\sum_{i=1}^{m} a_{ij}y_j \leq \sum_{i=1}^{n} a_{ij}x_i \leq \alpha\beta \sum_{i=1}^{m} a_{ij}y_j$

For the moment, we will only use $\alpha = 1$. We want to make primal conditions hold, then slacken the dual conditions.

Example Consider Algorithm 15.2 in the text:

---

1. Initialization: $x \leftarrow 0$; $y \leftarrow 0$
2. Until all elements are covered, do:

   Pick an uncovered element, say e, and raise y_e until some set goes tight.
   Pick all tight sets in the cover and update x.
   Declare all the elements occurring in these sets as "covered".
3. Output the set cover $\vec{x}$.

If f is the maximum frequency of any element, then $\sum_{S:e\in S} x_s \leq f$, so our dual slackness conditions hold with $\beta = f$ (see below). Note that the algorithm uses the primal/dual slackness conditions to achieve an approximation factor of f, but never actually solves the linear program. This is beneficial in many cases due to the ungainly complexity of linear program algorithms. Again, our LP is:

Primal:

$$\begin{array}{ll} \text{minimize} & \sum_{S\in\mathcal{S}} c(S)x_S \\ \text{subject to} & \sum_{S:e\in S} x_S \geq 1 \ \forall e \in \mathcal{U} \\ & x_s \geq 0 \ \forall S \in \mathcal{S} \end{array}$$

Dual:

$$\begin{array}{ll} \text{maximize} & \sum_{e\in\mathcal{U}} y_e \\ \text{subject to} & \sum_{e\in S} y_e \leq c(S) \ \forall S \in \mathcal{S} \\ & y_e \geq 0 \ \forall e \in \mathcal{U} \end{array}$$

We want to make the primal slackness conditions tight. The algorithm satisfies the relaxed complementary slackenss conditions with $\beta = f$ and $\alpha = 1$ to achieve an f-approximation. Note that Algorithm 2.17 in the text is the equivalent combinatorial algorithm for $f = 2$.

Fact If, in Algorithm 15.2, we change "Pick an uncovered element e and raise y_e until some set goes tight" to "For each uncovered element e, raise each y_e the same minimal amount such that a set goes tight", we obtain the "obvious" greedy set cover algorithm.

4 The Uncapacitated Facility Location (UFL) Problem

Definition

- F: set of facilities, $|F| = n$
- C: set of cities / clients / customers, $|C| = m$
- The i-th facility has a opening cost of f_i
- $\forall j \in C$, c_{ij} is the cost to connect j-th city to i-th facility
- x_i: indicate whether facility i is open
- u_{ij}: indicate if city j is connected to facility i
- In the disjoint model $F \cap C = \emptyset$
- In the complete model $F = C$

A polynomial sized LP formulation of the problem is:

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{n} x_i f_i + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} u_{ij} \\ \text{subject to} & u_{ij} \leq x_i \\ & \sum_{i=1}^{n} u_{ij} \geq 1, \forall j \\ & u_{ij} \geq 0 \\ & x_i \geq 0 \end{array}$$

In the corresponding IP formulation, we would need to restrict: $u_{ij} \in \{0, 1\}$ and $x_i \in \{0, 1\}$

In the metric case, we have: $c_{ij} \leq c_{i'j} + c_{i'j'} + c_{ij'}$

Algorithms for UFL

Note that the set cover problem can be viewed as a UFL problem, with unit opening cost and all connection cost $c_{ij} \in \{0, \infty\}$. Set S_i becomes a facility with $f_i = cost(S_i)$, and if $e_j \in S_i$ then $c_{ij} = 0$, else ∞.

The UFL problem can be viewed as a set cover problem as well. We introduce the notion of a *star* and cost-effectiveness. Let us say that a *star* consists of one facility and several cities. The cost of a star is the sum of the opening cost of the facility and the connection costs between the facility and all the cities in the star. More formally, the cost of the star (i, C'), where i is a facility and $C' \subseteq C$ is a subset of cities, is $f_i + \sum_{j \in C'} c_{ij}$. The cost effectiveness of the star (i, C') is the ratio of the cost of the star to the size of C', i.e. $(f_i + \sum_{j \in C'} c_{ij})/|C'|$. Also, notice that although the number of pairs of facilities and subsets of cities is exponentially large, in each iteration the most cost-effective pair can be found in polynomial time. For each facility i, we can sort the cities in increasing order of their connection cost to i. It can be easily seen that the most cost- effective pair will consist of a facility and a set, containing the first k cities in this order, for some k. So we have:

Algorithm 0:

We have the set of all the stars, and try to pick a minimum cost set of stars such that each city is in at least one star. The "obvious" greedy set cover algorithm can be applied, and an H_n approximation can be obtained. Note that this approximation ratio holds for arbitrary connection costs and since set cover can also be viewed as a special case of facility location with $\{0, \infty\}$ connections costs, we do not expect to be bale to get a better approximation ratio.

But we can do better in the metric case. Note that a facility can be chosen again after being opened, but its opening cost needs to be counted only once. So we can set f_i to zero after the first time the facility is picked by the algorithm. So in the following algorithm, as far as cities are concerned, every city j is removed from U, when connected to an open facility, and is not taken into consideration again.

Algorithm 1:

---

1. Let U be the set of unconnected cities. In the beginning, all cities are unconnected. i.e. $U := C$ and all the facilities are unopened.

2. while $U \neq \emptyset$:

- Among all stars, find the most cost-effective one, (i, C'), open facility i, if it is not already open, and connect all cities in C' to i.
- Set $f_i := 0, U := U \backslash C'$.

The idea of cost-effectiveness essentially stems from a similar notion in the greedy algorithm for the set cover problem. In that algorithm, the cost effectiveness of a set S is defined to be the cost of S over the number of uncovered elements in S. In each iteration, the algorithm picks the most cost-effective set until all elements are covered. The most cost-effective set can be found either by using direct computation, or by using the dual program of the linear programming formulation for the problem. The dual program can also be used to prove the approximation factor of the algorithm. Similarly, we will use the LP formulation of facility location to analyze our algorithm. As we will see, the dual formulation of the problem helps us to understand the nature of the problem and the greedy algorithm.

We give another LP formulation of the problem. Let $\mathcal{S}$ be the set of all stars. The facility location problem can be thought of as picking a minimum cost set of stars such that each city is in at least one star. this problem can be captured by the following integer program. Here x_S is an indicator denoting whether star S is picked and c_s denotes the cost of star S.

$$\begin{array}{ll} \text{Minimize} & \sum_{S \in \mathcal{S}} c_S x_S \\ \text{subject to} & \forall j \in C : \sum_{j \in S} x_S \geq 1 \\ & \forall S \in \mathcal{S} : x_S \in \{0, 1\} \end{array}$$

The LP-relaxation of this program is obtained by letting $\forall S \in \mathcal{S} : x_S \geq 0$

The dual program is:

$$\begin{array}{rl} \text{Maximize} & \sum_{j \in C} y_j \\ \text{subject to} & \forall S \in \mathcal{S} : \sum_{j \in s \cap C} y_j \leq c_S \\ & \forall j \in C : y_j \geq 0 \end{array}$$

There is an intuitive way of interpreting the dual variables. We can think of y_j as the contribution of city j, or its share toward the total expenses. Note that first inequality of the dual can also be written as $\sum_{j \in U} \max(0, y_j - c_{ij}) \leq f_i$ for every facility i. We can now see how the dual variables help us find the most cost-effective star in each iteration of the greedy algorithm. Hence we restate Algorithm 1 based on the above observation.

Restatement of Algorithm 1:

---

1. We introduce a notion of time, so that each event can be associated with the time at which it happened. The algorithm starts at time 0. Initially, each city is defined to be unconnected $(U := C)$, all facilities are unopened, and y_j is set to 0 for every j.

2. while $U \neq 0$, increase the time, and simultaneously, for every city $j \in U$, increase the parameter y_j at the same rate, until one of the following events occurs (if two events occur at the same time, we process them in arbitrary order).

(a) For some unconnected city j, and some open facility i, $y_j = c_{ij}$. In this case, connect city j to facility i and remove j from U.

(b) For some unopened facility i, we have $\sum_{j \in U} \max(0, y_j - c_{ij}) = f_i$. this means that the total contribution of the cities is sufficient to open facility i. In this case, open this facility, and for every unconnected city j with $y_j \geq c_{ij}$, connect j to i, and remove it from U.

If we call each facility opening and city connection an event, then by induction on the sequence of events, it can be shown that the above primal-dual algorithm is a reformulation of algorithm 1.

5 References

[1] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, Vijay V. Vazirani: A Greedy Facility Location Algorithm Analyzed Using Dual Fitting with factor-Revealing LP. Submitted to JACM.