CSC2420: Algorithm Design, Analysis and Theory Spring (or Winter for pessimists) 2017

Allan Borodin

February 6, 2017

Lecture 5

Announcements:

- Assignment 1 is due February 13, at the start of clsss.
  Please ask for any clarifications that are needed!
- I am reserving Thursdays 3-4 for an office hour but I welcome discussing the course whenever I am free.
- Can we arrange a 1 or 2 hour lecture outside of the usual time? I am here reading week but not the following week.

Todays agenda:

- Some recent progress of the makespan problem in the restricted machines model.
- Linear Programming Duality
  - Primal dual algorithms
  - dual fitting anaylsis
  - factor revealing LPs
- Begin randomized algorithms if time permits?

The restricted machine makespan problem

- The restricted machines model is a special case of the unrelated machines problem where for every job j, $p_{j,i} \in \{p_j, \infty\}$. Hence the LST 2-approximation applies.
- LST show that it is NP hard to do better than a 1.5 approximation for the restricted machines (and hence unrelated machines) problem.
- Shmoys shows that for the special case that $p_j \in \{1, 2\}$ that the problem can be solved in polynomial time.
- There is a relatively new (somewhat strange) result due to Svensson [2011]. He shows how to approximate the value of the optimum makespan to within a factor of $33/17 \approx 1.9413 < 2$. This is proven constructively by a local search algorithm satisfying the approximation. However, the local search is not shown to terminate in polynomial time.
- Note that if we could determine the optimal makespan value in polynomial time, then we can also find an optimal solution in polynomial time. How? **However, the same cannot be said when we are only have an "approximately optimal value".**

The special case of graph orientation

- Consider the special case when there are (at most) two allowable machines for each job. This is called the graph orientation problem.
- It turns out easier to reason about the LP rounding applied to the graph orientation problem for the given IP/LP but still the integrality gap is 2.
- A more refined IP/LP by Eveblendr, Krcal and Sgall [2008] achieves a 1.75 approximation for the graph orientation problem.
- Even for the case when each job can only be scheduled on at most 3 machines, beating the 2-approximation remains an open problem.

Some concluding remarks (for now) about LP rounding

- We will return later to more LP applications. There are some nice notes by Allan Jepson providing some of the geometric concepts underlying LP solutions. (Note: these slides are password protected but I will provide password in class.) http://www.cs.toronto.edu/ jepson/csc373/index2012.html
- There can be, of course, many different IP/LP formulations for a given problem. In particular, one often adds additional constraints so that the polytope of the LP solutions is smaller.
- For example, in the vertex cover LP, one could simply add constraints $x_i + x_j + x_k \geq 2$ for every triangle in the graph and more generally, constraints for every odd length cycle. (These inequalities do not essentially change the integrality gap.)
- Adding such constraints corresponds to one round of what is called the LS lift and project method.
- There are a number of lift and project methods. If you are interested, then consult our local expert Toni Pitassi.

Duality: See Vazirani and Shmoys/Williamson texts, and Williamson article

- For a primal maximization (resp. minimization) LP in standard form, the dual LP is a minimization (resp. maximization) LP in standard form.
- Specifically, if the primal $\mathcal{P}$ is:
  - Minimize $\mathbf{c} \cdot \mathbf{x}$
  - subject to $A_{m \times n} \cdot \mathbf{x} \geq \mathbf{b}$
  - $\mathbf{x} \geq 0$
- then the dual LP $\mathcal{D}$ with dual variables $\mathbf{y}$ is:
  - Maximize $\mathbf{b} \cdot \mathbf{y}$
  - subject to $A^{tr}_{n \times m} \cdot \mathbf{y} \leq \mathbf{c}$
  - $\mathbf{y} \geq 0$
- Note that the dual (resp. primal) variables are in correspondence to primal (resp. dual) constraints.
- If we consider the dual $\mathcal{D}$ as the primal then its dual is the original primal $\mathcal{P}$. That is, the dual of the dual is the primal.

An example: set cover

The vertex cover problem is a special case of the set cover problem in which the elements are the edges and the vertices are the sets, each set (ie vertex v) consisting of the edges adjacent to v.

The set cover problem as an IP/LP

minimize $\sum_j w_j x_j$
subject to $\sum_{j:u_i \in S_j} x_j \geq 1$ for all i; that is, for all $u_i \in U$
$x_j \in \{0,1\}$ (resp. $x_j \geq 0$)

The dual LP

maximize $\sum_i y_i$
subject to $\sum_{i:u_i \in S_j} y_i \leq w_j$ for all j
$y_i \geq 0$

If all the parameters in a standard form minimization (resp. maximization) problem are non negative, then the problem is called a covering (resp. packing) problem. Note that the set cover problem is a covering problem and its dual is a packing problem.

Duality Theory Overview

- An essential aspect of duality is that a finite optimal value to either the primal or the dual determines an optimal value to both.
- The relation between these two can sometimes be easy to interpret. However, the interpretation of the dual may not always be intuitively meaningful.
- Still, duality is very useful because the duality principle states that optimization problems may be viewed from either of two perspectives and this might be useful as the solution of the dual might be much easier to calculate than the solution of the primal.
- In some cases, the dual might provide additional insight as to how to round the LP solution to an integral solution.
- Moreover, the relation between the primal $\mathcal{P}$ and the dual $\mathcal{D}$ will lead to primal-Dual algorithms and to the so-called dual fiiting analysis.
- In what follows we will assume the primal is a minimization problem to simplify the exposition.

Strong and Weak Duality

Strong Duality

If x^* and y^* are (finite) optimal primal and resp. dual solutions, then $\mathcal{D}(\mathbf{y}^*) = \mathcal{P}(\mathbf{x}^*)$.

Note: Before it was known that solving LPs was in polynomial time, it was observed that strong duality proves that LP (as a decision problem) is in $\mathbf{NP} \cap \mathbf{co} - \mathbf{NP}$ which strongly suggested that LP was not NP-complete.

Weak Duality for a Minimization Problem

If $\mathbf{x}$ and $\mathbf{y}$ are primal and resp. dual solutions, then $\mathcal{D}(\mathbf{y}) \leq \mathcal{P}(\mathbf{x})$.

- Duality can be motivated by asking how one can verify that the minimum in the primal is at least some value z. To get witnesses, one can explore non-negative scaling factors (i.e. the dual variables) that can be used as multipliers in the constraints. The multipliers, however, must not violate the objective (i.e cause any multiplies of a primal variable to exceed the coefficient in the objective) we are trying to bound.

Motivating duality

Consider the motivating example in V. Vazirani's text:

Primal

minimize $7x_1 + x_2 + 5x_3$

subject to

- (1) $x_1 - x_2 + 3x_3 \geq 10$
- (2) $5x_1 + 2x_2 - x_3 \geq 6$
- $x_1, x_2, x_3 \geq 0$

Dual

maximize $10y_1 + 6y_2$

subject to

$$y_1 + 5y_2 \leq 7$$
$$-y_1 + 2y_2 \leq 1$$
$$3y_1 - y_2 \leq 5$$
$$y_1, y_2 \geq 0$$

Adding (1) and (2) and comparing the coefficient for each x_i, we have:

$7x_1 + x_2 + 5x_3 \geq (x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3) \geq 10 + 6 = 16$

Better yet,

$7x_1 + x_2 + 5x_3 \geq 2(x_1 - x_2 + 3x_3) + (5x_1 + 2x_2 - x_3) \geq 26$

For an upper bound, setting $(x_1, x_2, x_3) = (7/4, 0, 11/4)$

$7x_1 + x_2 + 5x_3 = 7 \cdot (7/4) + 1 \cdot 0 + 5 \cdot (11/4) = 26$

This proves that the optimal value for the primal and dual solution $(y_1, y_2) = (2, 1)$ must be 26.

Easy to prove weak duality

The proof for weak duality

$$\begin{aligned}
&\mathbf{b} \cdot \mathbf{y} = \textstyle\sum_{j=1}^{m} b_j y_j \\
&\leq \textstyle\sum_{j=1}^{m} (\sum_{i=1}^{n} A_{ji} x_i) y_j \\
&\leq \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} (A_{ji} y_j) x_i \\
&\leq \textstyle\sum_{i=1}^{n} c_i x_i = \mathbf{c} \cdot \mathbf{x}
\end{aligned}$$

Max flow-min Cut in terms of duality

- While the max flow problem can be naturally formulated as a LP, the natural formulation for min cut is as an IP. However, for this IP, it can be shown that the *extreme point solutions* (i.e. the vertices of the polyhedron defined by the constraints) are all integral $\{0,1\}$ in each coordinate. Moreover, there is a precise sense in which max flow and min cut can be viewed as dual problems. See Vazarani (section 12.2).
- In order to formulate max flow in standard LP form we reformulate the problem so that all flows (i.e. the LP variables) are non-negative. And to state the objective as a simple linear function (of the flows) we add an edge of infinite capacity from the terminal t to the source s and hence define a circulation problem.

The max flow LP

maximize $f_{t,s}$

subject to $f_{i,j} \leq c_{i,j}$ for all $(i,j) \in E$

$$\sum_{j:(j,i)\in E} f_{j,i} - \sum_{j:(i,j)\in E} f_{i,j} \leq 0 \quad \text{for all } i \in V$$

$$f_{i,j} \geq 0 \quad \text{for all } (i,j) \in E$$

Max flow-min cut duality continued

For the primal edge capacity constraints, introduce dual ("distance") variables $d_{i,j}$ and for the vertex flow conservation constraints, introduce dual ("potential") variables p_i.

The fractional min cut dual

minimize $\sum_{(i,j)\in E} c_{i,j} d_{i,j}$

subject to $\quad d_{i,j} - p_i + p_j \geq 0$

$\quad\quad\quad\quad\quad p_s - p_t \geq 1$

$\quad\quad\quad\quad\quad d_{i,j} \geq 0; p_i \geq 0$

- Now consider the IP restriction : $d_{i,j}, p_i \in \{0,1\}$ and let $\{(d^*_{i,j}, p^*_i)\}$ be an intergal optimum.
- The $\{0,1\}$ restriction and second constraint forces $p^*_s = 1; p^*_t = 0$.
- The IP optimum then defines a cut (S,T) with $S = \{i | p^*_i = 1\}$ and $T = \{i | p^*_i = 0\}$.
- Suppose (i,j) is in the cut, then $p^*_i = 1, p^*_j = 0$ which by the first constraint forces $d_{i,j} = 1$.
- The optimal $\{0,1\}$ IP solution (of the dual) defines a a min cut.

Solving the f-frequency set cover by a primal dual algorithm

- In the f-frequency set cover problem, each element is contained in at most f sets.
- Clearly, the vertex cover problem is an instance of the 2-frequency set cover.
- As in the vertex cover LP rounding, we can similarly solve the f-frequency cover problem by obtaining an optimal solution $\{x_j^*\}$ to the (primal) LP and then rounding to obtain $\bar{x}_j = 1$ iff $x_j^* \geq \frac{1}{f}$. This is, as noted before, a conceptually simple method but requires solving the LP.
- We know that for a minimization problem, any dual solution is a lower bound on any primal solution. One possible goal in a primal dual method for a minimization problem will be to maintain a fractional feasible dual solution and continue to try improve the dual solution. As dual constraints become tight we then set the corresponding primal variables.

Primal dual for f-frequency set cover continued

Suggestive lemma

Claim: Let $\{y_i^*\}$ be an optimal solution to the dual LP and let $\mathcal{C}' = \{S_j | \sum_{e_i \in S_j} y_i^* = w_j\}$. Then $\mathcal{C}'$ is a cover.

This suggests the following algorithm:

Primal dual algorithm for set cover

Set $y_i = 0$ for all i; $\mathcal{C}' := \varnothing$
While there exists an e_i not covered by $\mathcal{C}'$
  Increase the dual variables y_i until there is some $j : \sum_{\{k: e_i \in S_j\}} y_i = w_j$
  $\mathcal{C}' := \mathcal{C}' \cup \{S_j\}$
  Freeze the y_i associated with the newly covered e_i
End While

Theorem: Approximation bound for primal dual algorithm

The cover formed by tight constraints in the dual solution provides an f approximation for the f-frequency set cover problem.

Comments on the primal dual algorithm

- What is being shown is that the integral primal solution is within a factor of f of the dual solution which implies that the primal dual algorithm is an f-approximation algorithm for the f-frequency set cover problem.
- In fact, what is being shown is that the integraility gap of this IP/LP formulation for f-frequency set cover problem is at most f.
- In terms of implementation we would calculate the minimum ϵ needed to make some constraint tight so as to chose which primal variable to set. This ϵ could be 0 if a previous iteration had more than one constraint that becomes tight simultaneously. This ϵ would then be subtracted from w_j for j such that $e_i \in S_j$.

More comments on primal dual algorithms

- We have just seen an example of a basic form of the primal dual method for a minimization problem. Namely, we start with an infeasible integral primal solution and feasible (fractional) dual. (For a covering primal problem and dual packing problem, the initial dual solution can be the all zero solution.) Unsatisfied primal constraints suggest which dual constraints might be tightened and when one or more dual constraints become tight this determines which primal variable(s) to set.
- Some primal dual minimization algorithms extend this basic form by using a second (reverse delete) stage to achieve minimality. Some primal dual maximization algorithms use a reverse delete to enforce feasibility. There is some (for me not precise) relation between primal dual and local ratio alvgorithms (see Bar-Yehuda and Rawitz)
- **NOTE:** In the primal dual method we are not solving any LPs. Primal dual algorithms are viewed as "combinatorial algorithms" and in some cases they might even suggest an explicit greedy algorithm.

A primal dual algorithm with reverse delete : the weighted vertex feedback problem

The vertex feedback problem

Given a graph $G = (V, E)$, a feedback vertex set (FVS) F is a subset of vertices whose removal will make the resulting graph acyclic. That is, if $S = V - F$, then $G[S] = (S, E[S])$ is acyclic where $G[S]$ is the graph induced by S.

- The (weighted) feedback vertex set problem is to compute a miniumm size (weight) feedback vertex set.
- The problem (i.e. in its decision version) was one of Karp's original NP complete problems. It has application to circuit design and constraint satisfaction problems. It is as hard as vertex cover.
- An obvious IP for this problem would have the constraints $\sum_{v \in C} x_v \geq 1$ for every cycle C in the graph. Not only is this possibly an exponential size IP (which may or may not not be a problem), it is known that the integrality gap is $\Theta(\log |V|)$.

An alternative IP/LP for the FVS problem

- Chudak et al [1998] provide primal dual interpretations for the 2-approximation algorithms due to Becker and Geiger [1994] and Bafna, Berman, Fujito [1995]. In the primal dual interpretations, both algorithms use almost the same IP representation and method for raising dual variables.
- The basic fact underlying the IP representations is the following:

Fact

Let $d(v)$ be the degree of v, $b(S) = |E[S]| - |S| + 1$ and $\tau(S) =$ the size of a minimal feedback set for $G[S]$. Then if F is any FVS, and $E[S] \neq \varnothing$ then

1. $\sum_{v \in F}[d_S(v) - 1] \geq b(S)$ for all $S \subseteq V$ and hence
2. $\sum_{v \in F} d_S(v) \geq b(S) + \tau(S)$

Primal dual for FVS continued

The IP/LP and the resulting primal dual algorithm is a little easier to state for the Berger and Geiger algorithm but the analysis is perhaps a little simpler for the Bafna et al. algorithm. Here is the formulation for the Berger and Geiger algorithm:

Primal for Berger and Geiger algorithm

$\mathcal{P}$: minimize $\sum_{v \in V} w_v x_v$
subject to $\sum_{v \in S} d_S(v) x_v \geq b(S) + \tau(S)$ for all $S \subseteq V$ with $E[S] \neq \varnothing$
IP: $x_v \in \{0, 1\}$ LP: $x_v \geq 0$

The dual

$\mathcal{D}$: maximize $\sum_S (b(S) + \tau(S)) y_S$
subject to $\sum_{S : v \in S} d_S(v) y_S \leq w_v$ for all $v \in V$
$y_S \geq 0$ for all $S \subseteq V$ with $E[S] \neq \varnothing$

Note: These are exponential size LPs but that will not be a problem.

Primal dual for Berger and Geiger

$y_v = 0$ for all v; $\ell := 0$; $F := \emptyset$
$V' := V$; $E' := E$
While F is not a FVS for (V', E')
  $\ell := \ell + 1$
  recursively remove all isolated vertices and degree 1 vertices and incident edges from (V', E')
  $S := V'$ In the Bafna et al algorithm S is not always set to V'
  Increase y_S until $\exists v_\ell \in S$: $\sum_{T: v_\ell \in T} d_T(S) v_T = w_{v_\ell}$
  $F := F \cup \{v_\ell\}$
  Remove v_ℓ from V' and all incident edges from E'
End While
For $j = \ell..1$ % This is the reverse delete phase
  If $F - \{v_j\}$ is an FVS then $F := F - \{v_j\}$
  End If
End For

Comments on the primal dual for Berger and Geiger algorithm

- The algorithm as originally stated shows how to efficiently find a v_ℓ so as to make the the dual constraint tight; namely let $v_\ell = argmin_{v \in S} w_v / d_S(v_\ell)$ and let $\epsilon = w_{v_\ell} / d_S(v_\ell)$. Then $\epsilon d_S(u)$ is subtracted from w_u for all $u \in S$.
- It is easy to verify that any FVS is a solution to the primal and conversely any IP solution is an FVS.
- It is immediate that the F computed is an (integral) FVS since the **While** condition forces this.
- The analysis shows that for the dual LP constructs a feasible fractional $\{y_S\}$ solution satisfying: $\sum_{v \in F} w_v \leq 2 \sum_S (b(S) + \tau(S)) - 2 \sum_S y_S \leq 2 \sum_S (b(S) + \tau(S))$
- Therefore, the primal dual algorithm is a 2-approximation algorithm.
- The integrality gap is then at most 2 and this is known to be tight. It is also interesting to note that the dual objective function cannot be efficiently evaluated since $\tau(S)$ is the optimal FVS value for $G[S]$.

Using dual fitting to prove the approximation ratio of the greedy set cover algorithm

We have already seen the following natural greedy algorithm for the weighted set cover problem:

The greedy set cover algorithm

$\mathcal{C}' := \varnothing$
While there are uncovered elements
  Choose S_j such that $\frac{w_j}{|\tilde{S}_j|}$ is a minimum where
    $\tilde{S}_j$ is the subset of S_j containing the currently uncovered elements
  $\mathcal{C}' := \mathcal{C}' \cup S_j$
End While

We wish to prove the following theorem (Lovasz[1975], Chvatal [1979]):

Approximation ratio for greedy set cover

The approximation algorithm for the greedy algorithm is H_d where d is the maximum size of any set S_j.

The dual fitting analysis

The greedy set cover algorithm setting prices for each element

$\mathcal{C}' := \varnothing$

While there are uncovered elements

Choose S_j such that $\frac{w_j}{|\tilde{S}_j|}$ is a minimum where

$\tilde{S}_j$ is the subset of S_j containing the currently uncovered elements

%Charge each element e in $\tilde{S}_j$ the average cost $price(e) = \frac{w_j}{|\tilde{S}_j|}$

% This charging is just for the purpose of analysis

$\mathcal{C}' := \mathcal{C}' \cup S_j$

End While

- We can account for the cost of the solution by the costs imposed on the elements; namely, $\{price(e)\}$. That is, the cost of the greedy solution is $\sum_e price(e)$.

Dual fitting analysis continued

- The goal of the dual fitting analysis is to show that $y_e = price(e)/H_d$ is a feasible dual and hence any primal solution must have cost at least $\sum_e price(e)/H_d$.
- Consider any set $S = S_j$ in $\mathcal{C}$ having say $k \leq d$ elements. Let $e_1, \ldots, e_k$ be the elements of S in the order covered by the greedy algorithm (breaking ties arbitrarily). Consider the iteration is which e_i is first covered. At this iteration $\tilde{S}$ must have at least $k - i + 1$ uncovered elements and hence S could cover cover e_i at the average cost of $\frac{w_j}{k-i+1}$. Since the greedy algorithm chooses the most cost efficient set, $price(e_i) \leq \frac{w_j}{k-i+1}$.
- Summing over all elements in S_j, we have $\sum_{e_i \in S_j} y_{e_i} = \sum_{e_i \in S_j} price(e_i)/H_d \leq \sum_{e_i \in S_j} \frac{w_j}{k-i+1} \frac{1}{H_d} = w_j \frac{H_k}{H_d} \leq w_j$. Hence $\{y_e\}$ is a feasible dual.

The Steiner tree and Steiner forest problems

I am briefly presenting the treatment of this topic as in the "Approximation Algorithms" text by Vijay Vazirani (with slightly different notation. The development of primal dual algorithms began with paper by Agarwal, Klein and Ravi [1991] and Goemans and Williamson [1995] which provided a 2-approximation for the Steiner forest problem.

The Steiner forest problem on input $(G, c, \{R_i\})$ is defined as follows: We are given an edge weighted $c : E \to \mathbb{R}^{\geq 0}$ graph $G = (V, E)$ and disjoint subsets $R_1, \ldots, R_k$ of V. Let $S = V \setminus (\cup_i R_i)$. The nodes in $\cup_i R_i$ are called required nodes and the remaining nodes are called Steiner nodes. Without loss of generality, solutions will be forests.

The goal is to select a mininal cost set of edges so that the nodes in each R_i are connected. When $k = 1$, this is the Steiner tree problem
The Steiner tree problem has a relatively simple combinatorial 2-approximation algorithm. Agorithms with better approximation ratios are konwn. I believe the primal dual algorithm for the Steiner forest problem remains the best (poly time) approximation.

Combinatorial algorithm for Steiner tree

Without loss of generality, we can assume that the edge costs are a metric by the following metric closure:

- The graph G is transformed into a complete graph G' whose edge costs are a metric. Namely, replace each missing edge (u, v) by an edge with cost c' equal to the shortest cost between u and v.
- The optimum cost of a Steiner tree for (G, c', R) is at least as good as any solution for (G, c, R).
- Having found the optimal Steiner tree for (G, c', R), replace each edge in $E' \setminus E$ by the corresponding path. This will possibly create cycles so remove edges to form a tree T.

It sufices now to solve the Steiner tree problem for a metric edge cost c.

The metric Steiner tree problem

We solve the metric Steiner tree problem by computing a minimum cost spanning tree (MST) covering the nodes in R (and possibly using Steiner nodes).

Let OPT denote the cost of an optimal Steiner tree for (G, c, R). The cost of an MST on R is at most $2 \cdot OPT$

The proof follows by making two copies of edge edge which then makes the graph Eulerian (i.e. can be tranversed using each edge once). Then short cut the Steiner nodes and any previously traversed nodes to produce a Hamitonian cycle. These short cuts do not increase the cost for the Eulerian graph. The edge copies cause at most a factor of 2 in cost .

The bound for this algorithm is tight in that there is a class of graphs $\{G_n\}$ where G_n has n required nodes and one Steiner node. The MST for this graph has cost $2n - 1$ whereas the OPT cost is n.

The primal dual Steiner forest algorithm

Let $f(S) = 1$ if $u \in S, v \notin S$ for some u, v in the same required set R_i and 0 otherwise. Also let $\delta(S)$ denote the edges crossing the cut $(S, \bar{S})$. Then an IP formulation for the Steiner forest problem is as follows::

minimize $\sum_{e \in E} c_e x_e$
subject to $\sum_{e : e \in \delta(S)} x_e \geq f(S) \qquad S \subseteq V,$
$x_e \in \{0, 1\} \qquad e \in E.$

In the LP relaxation, $x_e \geq 0$ (dropping the unnecessary $x_e \leq 1$)

The dual is :

maximize $\sum_{S \subseteq V} f(S) y_S$
subject to $\sum_{S : e \in \delta(S)} y_S \leq c_e \qquad e \in E,$
$y_S \geq 0 \qquad S \subseteq V$

The Steiner forest primal dual

A set S is *unsatisfied* if $f(S) = 1$ but so far no chosen edge crossing the cut $(S, \bar{S})$ and a set is called *active* if it is a minimal (wrt. set inclusion) unsatisfied set. The current primal indicates which sets are unsatisfied and therefore which dual y_S variables have to be raised. This in turn will determine which edge will be chosen (i.e. when a dual constraint becomes tight).

Algorithm 22.3 (Steiner forest)

1. **(Initialization)** $F \leftarrow \emptyset$; for each $S \subseteq V$, $y_S \leftarrow 0$.
2. **(Edge augmentation)** while there exists an unsatisfied set do:
   simultaneously raise y_S for each active set S, until some edge e goes tight;
   $F \leftarrow F \cup \{e\}$.
3. **(Pruning)** return $F' = \{e \in F \mid F - \{e\}$ is primal infeasible$\}$

[From Vazirani text, Chapter 22]

Randomized algorithms

Our next theme will be randomized algorithms. For the main part, our previous themes have been on algorithmic paradigms. Randomization is not per se an algorithmic paradigm (in the same sense as greedy algorithms, DP, local search, LP rounding, primal dual algorithms).

Randomized algorithms

Our next theme will be randomized algorithms. For the main part, our previous themes have been on algorithmic paradigms. Randomization is not per se an algorithmic paradigm (in the same sense as greedy algorithms, DP, local search, LP rounding, primal dual algorithms).

Rather, randomization can be thought of as a tool that can be used in conjuction with any algorithmic paradigm. However, its use is so prominent and varied in algorithm design and analysis, that it takes on the sense of an algorithmic way of thinking.

The why of randomized algorithms

- There are some problem settings (e.g. simulation, cryptography, interactive proofs, sublinear time algorithms) where randomization is necessary.
- We can use randomization to improve approximation ratios.
- Even when a given algorithm can be derandomized, there is often conceptual insight to be gained from the initial randomized algorithm.
- In complexity theory a fundamental question is how much can randomization lower the time complexity of a problem. For decision problems, there are three polynomial time randomized classes ZPP (zero-sided), RP (1-sided) and BPP (2-sided) error. The big question (and conjecture?) is BPP $=$ P?
- One important aspect of randomized algorithms is that the probability of success can be amplified by repreated independent trials of the algorithm.

Some problems in randomized polynomial time not known to be in polynomial time

1. The symbolic determinant problem.
2. Given n, find a prime in $[2^n, 2^{n+1}]$
3. Estimating volume of a convex body given by a set of linear inequalitiies.
4. Solving a quadratic equation in $Z_p[x]$ for a large prime p.

Polynomial identity testing

- The general problem concerning polynomial identities is that we are implicitly given two multivariate polynomials and wish to determine if they are identical. One way we could be implicitly given these polynomials is by an arithmetic circuit. A specific case of interest is the following symbolic determinant problem.
- Consider an $n \times n$ matrix $A = (a_{i,j})$ whose entries are polynomials of total degree (at most) d in m variables, say with integer coeficients. The determinant $det(A) = \sum_{\pi \in S_n} (-1)^{sgn(\pi)} \prod_{i=1}^{n} a_{i,\pi(i)}$, is a polynomial of degree nd. The symbolic determinant problem is to determine whether $det(A) \equiv \mathbf{0}$, the zero polynomial.

Schwartz Zipple Lemma

Let $P \in \mathbf{F}[x_1, \ldots, x_m]$ be a non zero polynomial over a field $\mathbf{F}$ of total degree at most d. Let S be a finite subset of $\mathbf{F}$. Then $Prob_{r_i \in_u S}[P(r_1, \ldots .r_m) = 0] \leq \frac{d}{|S|}$

Schwartz Zipple is clearly a multivariate generalization of the fact that a univariate polynomial of degree d can have at most d zeros.

Polynomial identity testing and symbolic determinant continued

- Returning to the symbolic determinant problem, suppose then we choose a suffciently large set of integers S (for definiteness say $|S| \geq 2nd$). Randomly choosing $r_i \in S$, we evaluate each of the polynomial entries at the values $x_i = r_i$. We then have a matrix A' with (not so large) integer entries.
- We know how to compute the determinant of any such integer matrix $A'_{n \times n}$ in $O(n^3)$ arithmetic operations. (Using the currently fastest, but not necessarily practical, matrix multiplication algorithm the determinant can be computed in $O(n^{2.38})$ arithmetic operations.)
- That is, we are computing the $det(A)$ at random $r_i \in S$ which is a degree nd polynomial. Since $|S| \geq 2nd$, then $Prob[det(A') = 0] \leq \frac{1}{2}$ assuming $det(A) \not\equiv \mathbf{0}$. The probability of correctness con be amplifed by choosing a bigger S or by repeated trials.
- In complexity theory terms, the problem (is $det(A) \equiv \mathbf{0}$) is in co-RP.