CSC2420 Fall 2012: Algorithm Design, Analysis and Theory
Lecture 6

Allan Borodin

February 25, 2016

Announcements and todays agenda

- Announcements
  1. Assignment 1 is graded and being returned today. One person did not provide their name! The assignment was marked out of 90 and the average was $73/90 \approx 81\%$ and the median was $75/90 \approx 83$. Even if you did not do well, there is plenty of time to improve your grade.
  2. A couple of people wound up searching the internet and finally found a paper that solved the ROM question. I was totally surprised to find that someone actually wrote a paper on this one observation. BUT in terms of learning, this isn't the most productive way to spend your time.
  3. I had hoped to start Assignment 2 over reading week but didn't. I will post some initial questions by the weekend.
  4. There was a talk today relating to AGT (algorithmic game theory). Tomorrow he will be meeting with graduate students between 4 and 5:30 in the theory lab, SF 4302
  5. Tomorrow at 11, Mark Bun will be speaking at 11 in WB 119. The topic is differential privacy.
- Todays agenda
  1. Continue discussion of LP Duality
  2. Start randomized algorithms

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

As already noted, 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 $\sum_{j:e_i \in S_j} x_j \geq 1$ for all i; that is, $e_i \in U$
$x_j \in \{0, 1\}$ (resp. $x_j \geq 0$)

The dual LP

maximize $\sum_i y_i$
subject to $\sum_{i:e_i \in S_j} y_i \leq w_j$ 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 initially 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	Dual
minimize $7x_1 + x_2 + 5x_3$	maximize $10y_1 + 6y_2$
subject to	subject to
• (1) $x_1 - x_2 + 3x_3 \geq 10$	$y_1 + 5y_2 \leq 7$
• (2) $5x_1 + 2x_2 - x_3 \geq 6$	$-y_1 + 2y_2 \leq 1$
	$3y_1 - y_2 \leq 5$
• $x_1, x_2, x_3 \geq 0$	$y_1, y_2 \geq 0$

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 (with solution $(y_1, y_2) = (2, 1)$ must be 26.

Easy to prove weak duality

The proof for weak duality

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

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. This is described nicely in 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

$$\begin{aligned} \text{minimize } & \textstyle\sum_{(i,j)\in E} c_{i,j} d_{i,j} \\ \text{subject to } & d_{i,j} - p_i + p_j \geq 0 \\ & p_s - p_t \geq 1 \\ & d_{i,j} \geq 0; p_i \geq 0 \end{aligned}$$

- 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 $e_i \in S_j$.

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.

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 algorithms extend this basic form by using a second (reverse delete) stage to achieve minimality.
- **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.

Dual fitting applied to a maximization problem

Krysta [2005] applies dual fitting approach to a maximization problem, namely to analyze (in my terminology) fixed order prioriity algorithms (such as the Lehman et al [1999] greedy $2\sqrt{m}$ approximate set packing algorithm) for generalizations of the weighted set packing problem (which can be used to formulate many natural integer packing problems).

Generalized Set Packing

As in weighted set packing, we have a collection of sets $S \in \mathcal{S}$ over some universe U. Each set has a weight w_S. Now we allow sets to be multi-sets and let $q(u, S)$ to be the number of copies of $u \in S$. Furthermore, we also allow each element $u \in U$ to have some maximum number b_u of copies that can occur in a feasible solution (in contrast to the basic set packing problem where $b_u = 1$ for all $u \in U$).
The goal is to select a subcollection $\mathcal{C}$ of sets satisfying the feasibility constraints on the $\{b_u\}$ so as to maximize the sum of the weights of the sets in $\mathcal{C}$.

The natural IP and LP relaxation

The natural IP/LP

max $\sum_{S \in \mathcal{S}} w_S x_S$

- subject to $\sum_{S:u \in S} q(u,S) x_S \leq b_u \quad \forall u \in U$
- $x_S \in \{0,1\}$

In the LP relaxation, the {0,1} constraint becomes $0 \leq x_S \leq 1\}$
NOTE: Unlike set cover, for set packing the condition $x_S \leq 1$ is necessary

The minimization dual

min $\sum_{u \in U} b_u y_u + \sum_{S \in \mathcal{S}} z_S$

- subject to $z_S + \sum_{u \in S} q(u,S) y_u \geq w_S \quad \forall S \in \mathcal{S}$
- $z_S, y_u \geq 0$

NOTE: The dual variable z_S corresponds to the constraint $x_S \leq 1$

The secretary problem as an LP

We recall the classical secrtary problem (defined in Lecture 2) which is to maximize the probability of choosing the best candidate from N candidates that arrive in random order. Bucnbinder, Kain and Singh [2010] show how to view the classical secretary problem (and many generalization) as an LP maximization problem with the following benefits:

1. Finding an optimal mechanism reduces to solving a specific linear program
2. Proving that $\frac{1}{e}$ is the best bound possible reduces to finding a solution to the dual of the LP.
3. This approach facilitates the analysis of many generalizations of the secretary problem (i.e. by adding additional constraints or modifying the objective function).
4. One of the generalizations is to obtain a *truthful* mechanism whereby agents (i.e. candidates) have no incentive to seek a particular place in the ordering (and hence making a random order more meaningful).

The LP for the classical secretary problem

The primal LP $\mathcal{P}$

$\max \frac{1}{n} \sum_{i=1}^{N} i \cdot p_i$

- subject to: $i \cdot p_i \leq 1 - \sum_{j=1}^{i-1} p_j \quad 1 \leq i \leq N$
- $p_i \geq 0$

The dual LP $\mathcal{D}$

$\min \sum_{i}^{N} x_i$

- subject to: $\sum_{j=i+1}^{N} x_j + i \cdot x_i \geq \frac{i}{N} \quad 1 \leq i \leq N$
- $x_i \geq 0$

Sketch of LP characterization

To prove that this LP captures the secrtary problem one needs to prove:

- If M is any mechanism and p_i^M is the probability that M selects the candidate in position i. Then $\{p_i^M\}$ is a feasible solution for the primal $\mathcal{P}$ and Prob[M selects best candidate] $\leq$ the objective value of $\mathcal{P}$
- Let $\{p_i\}$ be any feasible solution of $\mathcal{P}$. Then the following mechanism M obtains the objective function of $\mathcal{P}$:
  Select candidate i with probability $\frac{i \cdot p_i}{(1-\sum_{j<i} p_j)}$ if the first $i-1$ candidates have not been selected and i is best so far.

Furthermore, to prove an upper bound (namely $\frac{1}{e} + o(1)$) on the best performance (i.e. best probability), it suffices to construct a feasible solution $\{x_i\}$ for the dual $\mathcal{D}$ with dual objective value $\frac{1}{e}$.

- Setting $x_i = 0$ for $1 \leq i \leq N/e$ and $x_i = \frac{1}{N}(1 - \sum_{j=i}^{N} \frac{1}{j})$ for $n/e < i \leq N$ is a feasible dual solution with value $\frac{1}{e}$.

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 algorithms extend this basic form by using a second (reverse delete) stage to achieve minimality.
- **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 might 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 $E[S] \neq \emptyset$
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 $E[S] \neq \emptyset$

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; $\ell := 0$; $F := \varnothing$
$V' := V$; $E' := E$
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 $G[S]$.

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 (jn 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 (jn 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.

The naive randomized algorithm for exact Max-k-Sat

We continue our discussion of randomized algorthms by considering the use of randomization for improving approximation algorithms. In this context, randomization can be (and is) combined with any type of algorithm.

Warning: For the following discussion of Max-Sat, we will follow the prevailing convention by stating approximation ratios as fractions $c < 1$.

- Consider the exact Max-k-Sat problem where we are given a CNF propositional formula in which every clause has exactly k literals. We consider the weighted case in which clauses have weights. The goal is to find a satisfying assignment that maximizes the size (or weight) of clauses that are satisfied.
- Since exact Max-k-Sat generalizes the exact k- SAT decision problem, it is clearly an NP hard problem for $k \geq 3$. It is interesting to note that while 2-SAT is polynomial time computable, Max-2-Sat is still NP hard.
- The naive randomized (online) algorithm for Max-k-Sat is to randomly set each variable to *true* or *false* with equal probability.

Analysis of naive Max-k-Sat algorithm continued

- Since the expectation of a sum is the sum of the expectations, we just have to consider the probability that a clause is satisfied to determine the expected weight of a clause.
- Since each clause C_i has k variables, the probability that a random assignment of the literals in C_i will set the clause to be satisfied is exactly $\frac{2^k-1}{2^k}$. Hence $\mathbf{E}$ [weight of satisfied clauses] $= \frac{2^k-1}{2^k} \sum_i w_i$
- Of course, this probability only improves if some clauses have more than k literals. It is the small clauses that are the limiting factor in this analysis.
- This is not only an approxination ratio but moreover a "totality ratio" in that the algorithms expected value is a factor $\frac{2^k-1}{2^k}$ of the sum of all clause weights whether satisfied or not.
- We can hope that when measuring against an optimal solution (and not the sum of all clause weights), small clauses might not be as problematic as they are in the above analysis of the naive algorithm.

Derandomizing the naive algorithm

We can derandomize the naive algorithm by what is called the method of conditional expectations. Let $F[x_1, \ldots, x_n]$ be an exact k CNF formula over n propositional variables $\{x_i\}$. For notational simplicity let $true = 1$ and $false = 0$ and let $w(F)|\tau$ denote the weighted sum of satisfied clauses given truth assignment τ.

- Let x_j be any variable. We express $\mathbf{E}[w(F)|_{x_i \in_u \{0,1\}}]$ as $\mathbf{E}[w(F)|_{x_i \in_u \{0,1\}} | x_j = 1] \cdot (1/2) + \mathbf{E}[w(F)|_{x_i \in_u \{0,1\}} | x_j = 0] \cdot (1/2)$
- This implies that one of the choices for x_j will yield an expectation at least as large as the overall expectation.
- It is easy to determine how to set x_j since we can calculate the expectation clause by clause.
- We can continue to do this for each variable and thus obtain a deterministic solution whose weight is at least the overall expected value of the naive randomized algorithm.
- NOTE: The derandomization can be done so as to achieve an online algorithm. Here the (online) input items are the propostional variables. What input representation is needed so that it fits (say) the priority formulation for an online algorithm?