

# CSC2420 Spring 2015: Lecture 8

Allan Borodin

March 5, 2015

# Announcements and today's agenda

- Assignment 2 due next week. I will try to answer questions today.
- I will gradually be constructing questions for assignment 3 which will now be the last assignment for the course.
- Slides relating to Derek Corneil's lectures (which I am calling lecture 7) are posted on the web page.
- Today's agenda
  - 1 Quick review of duality
  - 2 Dual fitting analysis of greedy algorithms
  - 3 The secretary problem as an LP using duality to argue  $\frac{1}{e}$  optimality.
  - 4 Factor revealing LPs
  - 5 Start randomized algorithms.

## Quick review of duality

- 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  $A \cdot \mathbf{x} \geq \mathbf{b}$
  - ▶  $\mathbf{x} \geq 0$
- then the dual LP  $\mathcal{D}$  with **dual variables**  $\mathbf{y}$  is:
  - ▶ Maximize  $\mathbf{b} \cdot \mathbf{y}$
  - ▶ subject to  $A^{tr} \cdot \mathbf{y} \leq \mathbf{c}$
  - ▶  $\mathbf{y} \geq 0$
- 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

$$\begin{aligned} & \text{minimize } \sum_j w_j x_j \\ & \text{subject to } \sum_{j:e_i \in S_j} x_j \geq 1 \quad \text{for all } i \\ & \quad \quad x_j \in \{0, 1\} \quad (\text{resp. } x_j \geq 0) \end{aligned}$$

### The dual LP

$$\begin{aligned} & \text{maximize } \sum_i y_i \\ & \text{subject to } \sum_{i:e_i \in S_j} y_i \leq w_j \quad \text{for all } j \\ & \quad \quad y_i \geq 0 \end{aligned}$$

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 fitting](#) analysis.
- In what follows we will usually 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  $\mathcal{D}(y^*) = \mathcal{P}(x^*)$ .

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  $\mathbf{NP} \cap \mathbf{co-NP}$  which strongly suggested that LP was not NP-complete.

## Weak Duality

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

- 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.

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

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

## The greedy set cover algorithm

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

**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

$C' := \emptyset$

**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

$C' := C' \cup S_j$

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

## 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 priority 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 secretary problem (defined in Lecture 2) which is to maximize the probability of choosing the best candidate from  $N$  candidates that arrive in random order. Bucninder, 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 $\mathcal{D}$

$$\min \sum_{i=1}^N x_i$$

- 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 secretary 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  $\text{Prob}[M \text{ selects best candidate}] \leq$  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  $\frac{1}{e}$ .

## Factor revealing LPs

- In the dual fitting method (that we illustrated with the natural greedy algorithm for set cover problem), the dual solution is not a feasible dual. But the dual solution appropriately scaled down is a a feasible dual. For the set cover problem, if  $d$  is the maximum size of any set, then  $H_d$  is a sufficient scaling factor. (This is dual complementary slackness.)
- Is there a principled way to think about deriving appropriate scaling factors so that dual solutions become feasible? This will be the goal of [factor revealing LPs](#).
- The greedy algorithm can be recast as a primal dual algorithm where the *price*( $e_i$ ) becomes the dual variable  $y_i$  associated with element  $e_i$ . These dual  $\{y_i\}$  variables are raised simultaneously and whenever a dual constraint becomes tight for a set  $S_j$ , all the dual variables in  $S_j$  are frozen (i.e. no longer raised) and withdraw their contribution from all other sets in which they occur. Then  $S_j$  is added to the cover.

## Factor revealing LPs continued

- This then has the nice interpretation of the dual variables paying for the sets in the cover.
- By renaming, let the order in which the dual variables are covered be  $e_1, e_2, \dots, e_m$ . By the uniform raising of the dual variables we then have  $y_1 \leq y_2 \leq \dots \leq y_m$ .
- Let us say that a  $k$  element set  $S_j$  is selected when  $i - 1$  of its elements have already been covered (and hence frozen). Then  $(k - i + 1)y_i \geq w_j$ .
- The goal then is to see what is the least scaling factor that can be used to insure dual feasibility.
- For a fixed size problem (i.e. fixing  $n$  and  $m$ , the number of sets and elements), we want to maximize over all sets and all instances of that size to reveal a satisfactory scaling factor.



## Factor revealing LPs continued

For the set cover greedy algorithm recast as a primal dual algorithm, we have the following factor revealing LP problem (for instances of a given size):

### Factor revealing LP for set cover greedy algorithm

Maximize  $\frac{\sum_{i=1}^k y_i}{w_S}$  over  $\{y_i\}$  and all sets  $S$  (noting that  $w_S$  is now considered a variable)

subject to

- $y_i \leq y_{i+1} \quad 1 \leq i \leq k-1$
- $(k-i+1)y_i \leq w_S \quad 1 \leq i \leq k-1$
- $y_i \geq 0 \quad 1 \leq i \leq k$
- $w_S \geq 1$

## Factor revealing LP conclusion

- For any fixed size, the factor revealing LP provides an appropriate scaling factor.
- One then needs to consider the supremum of these values as the instance size grows.
- The hope is that by inspection of some small cases that one can see determine an appropriate scaling factor for all instance sizes. That is, the approach provides guidance for an eventual human derived proof.
- Factor revealing LPs have been used in a number of algorithmic analyses. It was first explicitly presented by Jain et al [2003] for greedy algorithms for the facility location problem.
- It has been extended by Mahdian and Yan to [2011] to the KVV Ranking algorithm for bipartite matching in the ROM model. Their extension to **strongly factor revealing LPs** is such that any member of the family of factor revealing LPs can be used to establish an appropriate scaling factor.
- Another variant called **tradeoff revealing LPs** was used by Mehta et al [2015] to analyze a greedy algorithm for the the adwords problem.

## 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 conjunction 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 repeated 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 inequalities.
- 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 coefficients. The determinant  $\det(A) = \sum_{\pi \in S_n} (-1)^{\text{sgn}(\pi)} \prod_{i=1}^n a_{i,\pi(i)}$ , is a polynomial of degree  $nd$ . The symbolic determinant problem is to determine whether  $\det(A) \equiv \mathbf{0}$ , the zero polynomial.

### Schwartz Zippel Lemma

Let  $P \in \mathbf{F}[x_1, \dots, x_m]$  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

$$\text{Prob}_{r_i \in_u S}[P(r_1, \dots, r_m) = 0] \leq \frac{d}{|S|}$$

Schwartz Zippel 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 sufficiently 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  $\text{Prob}[\det(A') = 0] \leq \frac{1}{2}$  assuming  $\det(A) \neq 0$ . The probability of correctness can be amplified by choosing a bigger  $S$  or by repeated trials.
- In complexity theory terms, the problem (is  $\det(A) \equiv 0$ ) is in co-RP.



# The naive randomized algorithm for exact Max- $k$ -Sat

We continue our discussion of randomized algorithms 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 approximation ratio but moreover a “totality ratio” in that the algorithm’s 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  $F[x_1, \dots, x_n]$  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  $\tau$ .

- 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 iso as to achieve an online algorithm.

## (Exact) Max- $k$ -Sat

- For exact Max-2-Sat (resp. Max-3-Sat), the approximation (and totality) ratio is  $\frac{3}{4}$  (resp.  $\frac{7}{8}$ ).
- For  $k \geq 3$ , using PCPs (probabilistically checkable proofs), Hastad proves that it is NP-hard to improve upon the  $\frac{2^k-1}{2^k}$  approximation ratio for Max- $k$ -Sat.
- For Max-2-Sat, the  $\frac{3}{4}$  ratio can be improved (as we will see) by the use of semi-definite programming (SDP).
- The analysis for exact Max- $k$ -Sat clearly needed the fact that all clauses have at least  $k$  clauses. What bound does the naive online randomized algorithm or its derandomization obtain for (not exact) Max-2-Sat or arbitrary Max-Sat (when there can be unit clauses)?

# Johnson's Max-Sat Algorithm

## Johnson's [1974] algorithm

For all clauses  $C_i$ ,  $w'_i := w_i / (2^{|C_i|})$

Let  $L$  be the set of clauses in formula  $F$  and  $X$  the set of variables

**For**  $x \in X$  (or until  $L$  empty)

Let  $P = \{C_i \in L \text{ such that } x \text{ occurs positively}\}$

Let  $N = \{C_j \in L \text{ such that } x \text{ occurs negatively}\}$

**If**  $\sum_{C_i \in P} w'_i \geq \sum_{C_j \in N} w'_j$

$x := \text{true}; L := LP$

**For all**  $C_r \in N$ ,  $w'_r := 2w'_r$  **End For**

**Else**

$x := \text{false}; L := L - N$

**For all**  $C_r \in P$ ,  $w'_r := 2w'_r$  **End For**

**End If**

Delete  $x$  from  $X$

**End For**

# Johnson's algorithm is the derandomized algorithm

- Twenty years after Johnson's algorithm, Yannakakis [1994] presented the naive algorithm and showed that Johnson's algorithm is the derandomized naive algorithm.
- Yannakakis also observed that for arbitrary Max-Sat, the approximation of Johnson's algorithm is at best  $\frac{2}{3}$ . For example, consider the 2-CNF  $F = (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge \bar{y}$  when variable  $x$  is first set to true.
- Chen, Friesen, Zheng [1999] showed that Johnson's algorithm achieves approximation ratio  $\frac{2}{3}$  for arbitrary weighted Max-Sat.
- For arbitrary Max-Sat (resp. Max-2-Sat), the current best approximation ratio is .797 (resp. .931) using semi-definite programming and randomized rounding.