

# Notes on Approximation Algorithms

Bhargav Samineni

## Abstract

A collection of some notes on the design and analysis of approximation algorithms and approximation techniques. Based mainly off of [1, 2].

## 1 Introduction to LP-Duality

### 1.1 LP-Duality

For a minimization linear program in canonical form, the goal is to find a non-negative, rational vector  $x$  that minimizes a given linear objective function in  $x$  subject to some linear constraints on  $x$ . If there are  $n$  decision variables and  $m$  linear constraints, then  $x \in \mathbb{Q}^n$ , the coefficients of the linear objective function can be represented by a vector  $c \in \mathbb{Q}^n$ , the coefficients of the linear constraints can be represented by a matrix  $A = (a_{ij}) \in \mathbb{Q}^{m \times n}$ , and the values of the linear constraints can be represented by a vector  $b \in \mathbb{Q}^m$ .

#### Definition 1 (Primal and Dual)

Given a linear programming problem in canonical form denoted as  $(P)$ , we can induce a problem denoted by  $(D)$  with the following form:

$$\begin{array}{ll} (P) & \text{minimize} \quad \sum_{j=1}^n c_j x_j \\ & \text{subject to} \quad \sum_{j=1}^n a_{ij} x_j \geq b_i \quad i = 1, \dots, m \\ & \quad \quad \quad x_j \geq 0 \quad j = 1, \dots, n \\ (D) & \text{maximize} \quad \sum_{i=1}^m b_i y_i \\ & \text{subject to} \quad \sum_{i=1}^m a_{ij} y_i \leq c_j \quad j = 1, \dots, n \\ & \quad \quad \quad y_i \geq 0 \quad i = 1, \dots, m \end{array}$$

where  $a_{ij}$ ,  $b_i$ , and  $c_i$  are given rational numbers and  $y_i$  corresponds to the  $i$ th inequality of  $(P)$ .

#### Definition 2 (Primal)

The problem  $(P)$  is referred to as the *primal*.

#### Definition 3 (Dual)

The *dual* of the primal is problem  $(D)$ .

Every feasible solution for the dual serves as a lower bound on the optimal objective function value of the primal. The reverse also holds in that every feasible solution to the primal serves as an upper bound on the optimal objective function value of the dual.

**Theorem 1 (Weak Duality).** If  $x = (x_1, \dots, x_n)$  is a feasible solution to the LP  $(P)$  and  $y = (y_1, \dots, y_m)$  a feasible solution to the LP  $(D)$ , then  $\sum_{j=1}^n c_j x_j \geq \sum_{i=1}^m b_i y_i$ .

*Proof.*

$$\sum_{j=1}^n c_j x_j \geq \sum_{j=1}^n \left( \sum_{i=1}^m a_{ij} y_i \right) x_j = \sum_{i=1}^m \left( \sum_{j=1}^n a_{ij} x_j \right) y_i \geq \sum_{i=1}^m b_i y_i$$

■

As a consequence of [Theorem 1](#), if we find that there exist some feasible solutions to [\(P\)](#) and [\(D\)](#) that have matching objection function values, then these solutions must be optimal.

**Theorem 2** (Strong Duality/LP-Duality). *If the LPs [\(P\)](#) and [\(D\)](#) are both feasible, and  $x^* = (x_1^*, \dots, x_n^*)$  and  $y^* = (y_1^*, \dots, y_m^*)$  are optimal solutions to [\(P\)](#) and [\(D\)](#), respectively, then  $\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*$ .*

As a corollary of [Theorem 2](#), we find that optimal solutions to [\(P\)](#) and [\(D\)](#) must satisfy a set of conditions.

**Corollary 1** (Complementary Slack Conditions). *Let  $x$  and  $y$  be feasible solutions to [\(P\)](#) and [\(D\)](#), respectively. Then  $x$  and  $y$  are both optimal iff the following are satisfied:*

1. For each  $1 \leq j \leq n$ , either  $x_j = 0$  or  $\sum_{i=1}^m a_{ij} y_i = c_j$
2. For each  $1 \leq i \leq m$ , either  $y_i = 0$  or  $\sum_{j=1}^n a_{ij} x_j = b_i$ .

## 1.2 Algorithm Design Techniques

**Definition 4** (Extreme Point Solution)

Consider the polyhedron defining the set of feasible solutions to an LP. A feasible solution is called an *extreme point solution* if it is a vertex of the polyhedron.

**Theorem 3.** *For any objective function, there is an extreme point solution that is optimal.*

Combinatorial optimization problems can often be stated as integer programs (IP), which can be relaxed to an LP. For the case of NP-Hard problems, however, the polyhedron that defines the set of feasible solutions to the LP-relaxation may not have integer vertices. With the LP-relaxation, the goal is to find near optimal integral solutions.

One method to get an approximation algorithm is through *LP-rounding*. Rounding involves solving the LP-relaxation and then converting the fractional solution to an integral solution while ensuring that in the process the cost does not increase dramatically. The approximation ratio is given by comparing the cost of the integral and fractional solutions.

Another method is the *primal-dual schema*. Consider an LP-relaxation as the primal program. Under the scheme, an integral solution to the primal and a feasible solution to the dual are constructed iteratively. Any feasible solution to the dual also provides a lower bound on OPT. The approximation ratio is given by comparing the two solutions.

### 1.2.1 Integrality Gap

**Definition 5** (Integrality Gap)

Given an LP-relaxation for a minimization problem, let  $\text{OPT}_f(I)$  be the cost of an optimal fractional solution to an instance  $I$  of the problem. Similarly, let  $\text{OPT}(I)$  denote the cost of an optimal solution

to the original IP for an instance  $I$  of the problem. The *integrality gap* of the relaxation is

$$\sup_I \frac{\text{OPT}(I)}{\text{OPT}_f(I)}.$$

In the case of a maximization problem, the integrality gap is given by the infimum of the ratio.

When the integrality gap of an LP-relaxation is exactly 1, such a relaxation is called an *exact relaxation*. If the cost of a solution found by an algorithm is compared to the cost of an optimal fractional solution (or feasible dual solution), the best approximation factor one can hope to achieve is the integrality gap of the relaxation.

### 1.3 Set Cover

#### Problem 1 (Set Cover)

Given a ground set  $U = \{e_1, \dots, e_n\}$ , a collection  $\mathcal{S} = \{S_1, \dots, S_m\}$  of subsets of  $U$ , and a cost function  $c: \mathcal{S} \rightarrow \mathbb{Q}^+$ , find a minimum cost subcollection of  $\mathcal{S}$  that covers all elements in  $U$ .

To formulate **Set Cover** as an IP, denote the binary decision variables by  $x_S$  for each  $S \in \mathcal{S}$ . If  $S$  is in the cover, then  $x_S = 1$ .

$$\begin{aligned} & \text{minimize} && \sum_{S \in \mathcal{S}} c(S)x_S \\ & \text{subject to} && \sum_{S: e \in S} x_S \geq 1 \quad e \in U \\ & && x_S \in \{0, 1\} \quad S \in \mathcal{S} \end{aligned} \tag{1}$$

The LP-relaxation to this IP is given by letting  $x_S \geq 0$ . A solution to the LP-relaxation can be viewed as a fractional set cover.

$$\begin{aligned} & \text{minimize} && \sum_{S \in \mathcal{S}} c(S)x_S \\ & \text{subject to} && \sum_{S: e \in S} x_S \geq 1 \quad e \in U \\ & && x_S \geq 0 \quad S \in \mathcal{S} \end{aligned} \tag{2}$$

Introducing a variable  $y_e$  for each  $e \in U$ , we get the dual program.

$$\begin{aligned} & \text{maximize} && \sum_{e \in U} y_e \\ & \text{subject to} && \sum_{e: e \in S} y_e \leq c(S) \quad S \in \mathcal{S} \\ & && y_e \geq 0 \quad e \in U \end{aligned} \tag{3}$$

---

#### Algorithm 1 Greedy Set Cover Algorithm

---

```

1:  $C \leftarrow \emptyset$ 
2: while  $C \neq U$  do
3:   Let  $\alpha_i = \frac{c(S_i)}{|S_i \setminus C|}$  be the cost effectiveness of a set  $S_i \in \mathcal{S}$ .
4:   Find the set that minimizes  $\alpha_i$ . Denote this set by  $S_j$  and set  $p(e) = \alpha_j$  for each  $e \in S_j \setminus C$ 
5:    $C \leftarrow C \cup S_j$ ,  $\mathcal{S} \leftarrow \mathcal{S} \setminus S_j$ 
6: return  $C$ 

```

---

### 1.3.1 Dual-Fitting Analysis

## 1.4 Exercises

### Exercise 1.1

*The dual of the dual is the primal.*

*Solution.* The dual (D) is equivalent to the minimization problem

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m -b_i y_i \\ & \text{subject to} && \sum_{i=1}^m -a_{ij} y_i \geq -c_j \quad j = 1, \dots, n \\ & && y_i \geq 0 \quad i = 1, \dots, m. \end{aligned}$$

Taking the dual of this LP gives

$$\begin{aligned} & \text{maximize} && \sum_{j=1}^n -c_j x_j \\ & \text{subject to} && \sum_{j=1}^n -a_{ij} x_j \leq -b_i \quad i = 1, \dots, m \\ & && x_j \geq 0 \quad j = 1, \dots, n \end{aligned} \iff \begin{aligned} & \text{minimize} && \sum_{j=1}^n c_j x_j \\ & \text{subject to} && \sum_{j=1}^n a_{ij} x_j \geq b_i \quad i = 1, \dots, m \\ & && x_j \geq 0 \quad j = 1, \dots, n \end{aligned}$$

which is the primal (P). ■

## References

- [1] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.
- [2] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.