Linear Programming Lecture Summary

Shalev David 036764751

Introduction

Linear Programming is a widely investigated Optimization Problem that can be solved in Polynomial Time and used in economics and engineering. In this work we will undergo the main ideas and outcomes of LP research.

At first we will present the definition of LP Problem and demonstrate some real-world problems that can be modeled as LPP's (LP Problems). Afterwards we will discuss the issue of solving LP, in particular state that this is a Language in P, break down to details its feasible region convexity significance for the Simplex method. Moreover, in context of Simplex method we will discuss its complexity Analysis according to worst-case, average-case, and smoothed complexity criterions.

In addition, we will discuss Integer Programming problem, prove it belongs to NPC, and show an approach to approximate it using LP relaxation.

At the end, we will discuss LP duality, which is the issue of transforming a maximum LP problem called primal into a minimum LP problem called dual. We will explain its Economic meaning, and specify the connections and equivalence between the primal-dual pair for its various possibilities.

LP definition

Linear programming is an optimization problem of the form $\begin{aligned} \text{Maximize} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0 \end{aligned}$

Where x is an n-dimensional vector and A is an nxm matrix. The goal is to maximize the objective linear function with cost coefficient n-dimensional vector c.

An LPP can be transformed into this kind of matrix form using some algebraic tricks for cases where the objective is to minimize or where the constraints are of other forms such as "greater or equal to", variables are non-positive/unrestricted to sign etc.

Knowledge of these algebraic tricks are important for pre-algebraic formulation and analysis, and for the important practical purpose of feeding A,b,c as input into some software package solver.

LP is an optimization problem widely used. Many real world problems lend themselves to linear programming modeling or alternatively can be approximated by linear models.

There are well-known successful applications in fields such as Manufacturing, Marketing, Finance (investment), Advertising and Agriculture.

In class, two problems from real world were modeled as LP:

1. Production Manufacturing for maximizing total profit, under resource and marketing constraints.

2. Placing a machine within a given machine layout, with the goal of minimizing total street distance. This was an out of box modeling through which an absolute 2D function with no constraint, was turned into 8D LPP with appropriate constraints.

Solving LP

In this section mathematical issues are covered for the purpose of introduction to algorithms and complexity analysis of Linear Programming.

It is proven that the optimum value of linear objective function over a convex polytope feasible region is at the vertex, and in addition a local optimum along a set of adjacent vertexes is also global thanks to the these assumptions (objective function linearity + feasible region convexity). The significance of its feasible region convexity and objective function linearity are fundamental for understanding the simplex method and for an initial intuition that this is a language in P.

Thanks to these characteristics LP was shown to be a language in P. Two known algorithms that solve LP in proven polynomial-time worst-case analysis are the interior-point method and the ellipsoid method.

However, the most dominant, popular and known algorithm for solving LP is the simplex method. This algorithm in practice is very fast, even though its worst-case analysis turned out to be Exponential (Klee-minty construction). The simplex method is a Two-Phase method. 1st phase focuses on finding an initial vertex on convex polytope feasible set. We explained in class that this is indeed **not any** arbitrary n constraints intersection solution. 2nd phase focuses on running greedily to adjacent vertexes for which the function increase slope is the largest. We mentioned that this algorithm always terminates (regardless of special case degeneracy treatment) and practically very fast.

LPs time complexity in both criterions average-case and smoothed-analysis are polynomial. The first and important average-case criterion argues that averaging time complexity over all instances of LP Language, is polynomial. The second criterion of smoothed analysis states the measurement of maximum over all LP average-case instances neighborhoods. For LP this measurement is polynomial-time which means completes the conclusion that indeed we should be very unlucky to find a bad LPP instance.

Integer Programming

Integer Programming Problem (IPP) is an LP problem with additional constraint that all variables are of integers values. Similarly, BIP is Binary Integer Programming where all variables can be assigned Binary values $\{0, 1\}$.

These families of problems can model many real world problems where a legal or applicable solution cannot be assign any real value, or where problem can be modeled only be defining variables with integral values.

IP and BIP are both NP complete languages. We prove so for BIP by showing that BIP $\in$NP and in addition a building a polynomial time reduction from the known NPC Vertex-Cover to BIP.

A preliminary step is to formulate the decision problem of BIP and VC as we recall that the analysis is performed on decision problems representation.

1. BIP $\in$NP: Given a certificate a verifier would check in polynomial time that all constraints are satisfied and the sum is at most K.

2. VC$\leq_p$BIP: the reduction for each vertex v $\in$ V we define a corresponding variable x_v —if v in the vertex cover then x_v equals 1, otherwise 0. We get a BIP Decision problem of the form:

$$\begin{aligned} &\text{Does } \langle G,k\rangle \text{ Satisfy } \Sigma x_v \le K \\ &\text{Subject to:} \quad x_v + x_u \ge 1 \text{ for every edge } (u,v)\in V \\ &\qquad\qquad x_{v\in\{0,1\}} \text{ for every } v\in V \end{aligned}$$

Firstly, in relation to the size of <G,K>, which is of order nxm due to graph representation (n vertexes and m edges), the reduction is done in polynomial time.

So now, if our instance w=<G,k> is a Vertex Cover (positive answer for VC decision problem), then we mapped it to an instance f(w) that is indeed a BIP (positive answer for BIP decision problem). If w is not a VC (negative answer for VC decision problem) then our instance f(w) is not a BIP (negative answer for BIP decision problem).

Eventually, the important conclusion is that Integer Programming problem is hard to solve. So if we were able to model a problem to IP what can we do next to solve it for a general instance. There are some useful approaches for IPP's approximations, such as Branch and Bound, or LP relaxation. Here we will explain LP relaxation approach for further details and specification.

LP relaxation approximation consists of several stages. Firstly, the LP relaxation, which is the immediate process of neglecting the integer constraints added, Hence returning to an LPP.

For any IPP it is clear that the solution of its corresponding LP relaxation optimal solution value OPT(LP) is at least not worse than OPT(IP). This is due to constraint relaxation (IPP's feasible set is included in the equivalent LPP's relaxed feasible set).

Now, a solution of the LP relaxed is done in polynomial time, but this solution will probably not be included in the original IP non-convex feasible set. Only linear constraints are kept satisfied in the solution, Integer constrains probably not.

Therefore, we should perform an appropriate rounding on the solution in a way that linear constraints are kept satisfied. Afterwards, approximation analysis is done in order to prove additive or constant factor approximation boundary.

In class we have taken minimum Vertex Cover Problem for example, converted it to BIP (minimum problem), performed LP relaxation and showed a rounding technique for which all constraints are satisfied and for which a 2-factor approximation is guaranteed.

LP Duality

An LP problem of the form $\max\{c^T x \mid Ax \leq b, x \geq 0\}$ is defined in literature as LP primal. For every such problem we define its dual to be of the form $\min\{b^T y \mid A^T y \geq c, y \geq 0\}$.

At first, if we examine the actions done in the process of building the Dual out of its primal, we come to a profound understanding of behavior in real world economics.

Given an LP Primal the technical actions for building its Dual is replacing between the variables and the constraints. So, x role is the amount of products sold with an objective is to maximize the total profit limited to resources upper bound. On the second point of view, y role is the cost of each resource with an objective to minimize total cost when it is assumed that the resource producer is aware of the profit made per piece (by manufacturer) and therefore charges at least this value.

Primal-Dual pair has very interesting and useful Relations between them. According to weak duality theory if x and y are feasible solutions to the Primal and Dual respectively, then value of Primal is never higher than Dual value. This relation was proven in class.

In addition, according to strong duality if Primal is feasible and has a finite optimum then Dual is feasible and has a finite optimum, and their optimal values are the same. This is also true on the opposite direction.

Acronym:

NP = **N**ondeterministic **P**olynomial

NPC = **NP C**omplete

NPH = **NP H**ard

LP = **L**inear **P**rogramming

IP = **I**nteger **P**rogramming

LPP = **L**inear **P**rogramming **P**roblem

IPP = **I**nteger **P**rogramming **P**roblem