Linear Programming

From slides by Paul Beame

The process of minimizing a linear objective function subject to a finite number of linear equality and inequality constraints.

The word “programming” is historical and predates computer programming.

Example applications:
- airline crew scheduling
- manufacturing and production planning
- telecommunications network design
- "Few problems studied in computer science have greater application in the real world."

Suggested Readings:
- Chapter 7 of text by Dasgupta, Papadimitriou, Vazirani (link on web page).
- "Linear Programming", by Howard Karloff
  - First 34 pages on Simplex Algorithm available through Google books preview
- "Linear Programming", by Vasek Chvatal
- "Understanding and Using Linear Programming", by Jiri Matousek and Bernd Gartner

An Example: The Diet Problem

A student is trying to decide on lowest cost diet that provides sufficient amount of protein, with two choices:
- steak: 2 units of protein/pound, $3/pound
- peanut butter: 1 unit of protein/pound, $2/pound

In proper diet, need 4 units protein/day.

Let \( x \) = # pounds peanut butter/day in the diet.
Let \( y \) = # pounds steak/day in the diet.

Goal: minimize \( 2x + 3y \) (total cost)
subject to constraints:
\[
\begin{align*}
x + 2y &\geq 4 \\
x &\geq 0, \ y &\geq 0
\end{align*}
\]

This is an LP- formulation of our problem
An Example: The Diet Problem

Goal: minimize $2x + 3y$ (total cost)
subject to constraints:
$x + 2y \geq 4$
$x \geq 0, \ y \geq 0$

- This is an optimization problem.
- Any solution meeting the nutritional demands is called a feasible solution.
- A feasible solution of minimum cost is called the optimal solution.

Linear Program - Definition

A linear program is a problem with $n$ variables $x_1, \ldots, x_n$, that has:
1. A linear objective function, which must be minimized/maximized. Looks like:
   $$\min (\max) \ c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$
2. A set of $m$ linear constraints. A constraint looks like:
   $$a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n \leq b_i \ (or \ \geq \ or \ =)$$

Note: the values of the coefficients $c_i, b_i, a_{ij}$ are given in the problem input.
Optimal vector occurs at some corner of the feasible set.

Standard Form of a Linear Program.

Maximize \( c_1x_1 + c_2x_2 + \ldots + c_nx_n \)
subject to \( \sum_{1 \leq j \leq n} a_{ij}x_j \leq b_j \) \( \quad i=1..m \)
\( x_j \geq 0 \) \( \quad j=1..n \)

or

Minimize \( b_1y_1 + b_2y_2 + \ldots + b_my_m \)
subject to \( \sum_{1 \leq i \leq m} a_{ij}y_i \geq c_j \) \( \quad j=1..n \)
\( y_i \geq 0 \) \( \quad i=1..m \)

Feasible Set

- Each linear inequality divides \( n \)-dimensional space into two halfspaces, one where the inequality is satisfied, and one where it's not.
- **Feasible Set**: solutions to a family of linear inequalities.
  - Convex: for any 2 points in feasible set, the line segment joining them is in feasible set.
- The linear cost functions, defines a family of parallel hyperplanes (lines in 2D, planes in 3D, etc.). Want to find one of minimum cost \( \Rightarrow \) must occur at corner of feasible set.
  - Corner= can't be expressed as convex combination of 2 or more points in feasible set.

The Feasible Set

- Intersection of a set of half-spaces, called a polyhedron.
  - If it's bounded and nonempty, it's a polytope.
- There are 3 cases:
  - feasible set is empty.
  - cost function is unbounded on feasible set.
  - cost has a minimum (or maximum) on feasible set.
First two cases very uncommon for real problems in economics, science and engineering.
Solving LPs

- There are several algorithms that solve any linear program optimally.
  - The Simplex method (class of methods, usually very good but worst-case exponential for known methods)
  - The Ellipsoid method (polynomial-time)
- More
- These algorithms can be implemented in various ways.
- There are many existing software packages for LP.
- It is convenient to use LP as a "black box" for solving various optimization problems.

LP formulation: another example

Bob's bakery sells bagel and muffins.
To bake a dozen bagels Bob needs 5 cups of flour, 2 eggs, and 1 cup of sugar.
To bake a dozen muffins Bob needs 4 cups of flour, 4 eggs and 2 cups of sugar.
Bob can sell bagels at $10/dozen and muffins at $12/dozen.
Bob has 50 cups of flour, 30 eggs and 20 cups of sugar.
How many bagels and muffins should Bob bake in order to maximize his revenue?

LP formulation: Bob's bakery

<table>
<thead>
<tr>
<th>Bagels</th>
<th>Muffins</th>
<th>Avail.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flour</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Eggs</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Sugar</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Maximize \( 10x_1 + 12x_2 \)
\[
\begin{align*}
\text{s.t.} & \quad 5x_1 + 4x_2 \leq 50 \\
& \quad 2x_1 + 4x_2 \leq 30 \\
& \quad x_1 + 2x_2 \leq 20 \\
& \quad x_1 \geq 0, x_2 \geq 0
\end{align*}
\]

Maximize \( c^T x \)
\[
\begin{align*}
\text{s.t.} & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*}
\]

Idea of the Simplex Method

The Toy Factory Problem (TFP):
A toy factory produces dolls and cars.
Danny, a new employee, is hired. He can produce 2 cars and 3 dolls a day. However, the packaging machine can only pack 4 items a day. The company’s profit from each doll is $10 and from each car is $15. What should Danny be asked to do?

Step 1: Describe the problem as an LP problem.
Let \( x_1, x_2 \) denote the number of cars and dolls produced by Danny.

Maximize \( 10x_1 + 15x_2 \)
\[
\begin{align*}
\text{s.t.} & \quad 2x_1 + 3x_2 \leq 4 \\
& \quad 5x_1 + 2x_2 \leq 4 \\
& \quad x_1 \geq 0, x_2 \geq 0
\end{align*}
\]
The Toy Factory Problem

Let \( x_1, x_2 \) denote the number of cars and dolls produced by Danny.

Objective:
Max \( z = 15x_1 + 10x_2 \)

s.t
\[
\begin{align*}
x_1 & \leq 2 \\
x_2 & \leq 3 \\
x_1 + x_2 & \leq 4 \\
x_1 & \geq 0 \\
x_2 & \geq 0
\end{align*}
\]

Important Observations:

1. An optimum solution to the LP is always at a corner point.

It might be that the objective line is parallel to a constraint. In this case there are many optimum points, in particular at the relevant corner points (consider \( z = 15x_1 + 15x_2 \)).

Important Observations:

2. If a corner point feasible solution has an objective function value that is better than or equal to all its adjacent corner point feasible solutions then it is optimal.

3. There is a finite number of corner point feasible solutions.
The Simplex Method

Phase 1 (start-up): Find any corner point feasible solution. In many standard LPs the origin can serve as the start-up corner point.

Phase 2 (iterate): Repeatedly move to a better adjacent corner point feasible solution until no further better adjacent corner point feasible solution can be found. The final corner point defines the optimum point.

Example: The Toy Factory Problem

Phase 1: start at (0,0)
Objective value = \( Z(0,0) = 0 \)

Iteration 1: Move to (2,0).
\( Z(2,0) = 30 \). An Improvement

Iteration 2: Move to (2,2)
\( Z(2,2) = 50 \). An Improvement

Iteration 3: Consider moving to (1,3), \( Z(1,3) = 45 < 50 \).
Conclude that (2,2) is optimum!

A Central Result of LP Theory: Duality Theorem

- Every linear program has a dual
- If the original is a minimization, the dual is a maximization and vice versa
- Solution of one leads to solution of other

Primal: Maximize \( c^T x \) subject to \( Ax \leq b \), \( x \geq 0 \)
Dual: Minimize \( b^T y \) subject to \( A^T y \geq c \), \( y \geq 0 \)

If one has optimal solution so does the other, and their values are the same.
**Simple Example**

- Diet problem: \( \text{minimize } 2x + 3y \)
  subject to \( x + 2y \geq 4, \ x \geq 0, \ y \geq 0 \)
- Dual problem: \( \text{maximize } 4p \)
  subject to \( p \leq 2, \ 2p \leq 3, \ p \geq 0 \)
- Dual: the problem faced by a druggist who sells synthetic protein, trying to compete with peanut butter and steak

**Proof of Weak Duality**

- Suppose that
  - \( x \) satisfies \( Ax \leq b, \ x \geq 0 \)
  - \( y \) satisfies \( A^T y \geq c, \ y \geq 0 \)
- Then
  - \( c^T x \leq (A^T y)^T x \) since \( x \geq 0 \) and \( A^T y \geq c \)
  - \( = y^T A x \) by definition
  - \( \leq y^T b \) since \( y \geq 0 \) and \( Ax \leq b \)
  - \( = b^T y \) by definition
- This says that any feasible solution to the primal (maximization problem) has an objective function value at most that of any feasible solution of the dual (minimization) problem.
- Strong duality says that the optima of the two are equal

**What’s going on?**

- Notice: feasible sets completely different for primal and dual, but nonetheless an important relation between them.
- Duality theorem says that in the competition between the grocer and the druggist the result is always a tie.
- Optimal solution to primal tells purchaser what to do.
- Optimal solution to dual fixes the natural prices at which economy should run.
- The diet \( x \) and vitamin prices \( y \) are optimal when
  - grocer sells zero of any food that is priced above its vitamin equivalent.
  - druggist charges 0 for any vitamin that is oversupplied in the diet.
Duality Theorem

Druggist’s max revenue = Purchasers min cost

Practical Use of Duality:
- Sometimes simplex algorithm (or other algorithms) will run faster on the dual than on the primal.
- Can be used to bound how far you are from optimal solution.
- Is used in algorithm design.
- Important implications for economists.

Example: Max Flow

Variables: $f_{uv}$ - the flow on edge $e=(u,v)$.

$$\text{Max } \sum u f_{su}$$

s.t.

$$f_{uv} \leq c_{uv}, \forall (u,v) \in E$$

$$\sum_u f_{uv} - \sum_v f_{vw} = 0, \forall v \in V-\{s,t\}$$

$$f_{uv} \geq 0, \forall (u,v) \in E$$

Ellipsoid Algorithm

- Running time is polynomial but depends on the # of bits $L$ needed to represent numbers in $A$, $b$, and $c$
- Idea: Hunt lion in Sahara (under assumption there is at most one)
  - Fence Sahara in
  - Divide in 2 halves with another fence
  - Detect one half that has no lion.
  - Continue recursively on other side until fenced area so small that either find lion or can argue that no lion could fit in there.

- In ellipsoid algorithm:
  - Fenced area is ellipsoid
  - Solve feasibility problem: does there exist $x$ s.t. $Ax \leq b$?

Ellipsoid Algorithm

- Running time is polynomial but depends on the # of bits $L$ needed to represent numbers in $A$, $b$, and $c$
  - Like capacity-scaling for network flow but a much bigger polynomial
  - Interior point methods running times also depend on $L$

- Method applies to large class of convex programs
  - Can be efficient for LPs with exponentially many constraints

- Open whether a strongly polynomial-time algorithm exists for LP
  - One where running time has # of operations polynomial in just $m$ and $n$
Integer Programming (IP)

- An LP problem with an additional requirement that variables will only get an integral value, maybe from some range.
- 01P – binary integer programming: variables should be assigned only 0 or 1.
- Can model many problems.
- NP-hard to solve!

01P Example: Vertex Cover

Variables: for each \( v \in V \), \( x_v \) – is \( v \) in the cover?
Minimize \( \sum x_v \)
Subject to: \( x_v \in \{0,1\} \)
\[ x_u + x_v \geq 1 \quad \forall (u,v) \in E \]

01P Example: Weighted Set Cover

Input: a Collection \( S_1, S_2, \ldots, S_n \) of subsets of \( \{1,2,3, \ldots, m\} \) a cost \( p_i \) for set \( S_i \).
Output: A collection of subsets whose union is \( \{1,2, \ldots, m\} \).
Objective: Minimum total cost of selected subsets.

Variables: For each subset, \( x_i \) – is subset \( S_i \) selected for the cover?
Minimize \( \sum p_i \cdot x_i \)
Subject to: \( x_i \in \{0,1\}^n \)
\[ \sum_{j \in S_i} x_j \geq 1 \quad \forall j = 1..m \]

01P Example: Shortest Path

Given a directed graph \( G(V,E) \), \( s, t \in V \) and length \( p_e \) for edge \( e \).

Variables: For each edge, \( x_e \) – is \( e \) in the path?
Minimize \( \sum p_e \cdot x_e \)
Subject to: \( x_e \in \{0,1\} \quad \forall e \in E \)
\[ \sum_{e \in A} x_e \geq 1 \quad \forall s \rightarrow t \text{ cut } A \]
LP-based approximations

- We don't know any polynomial-time algorithm for any NP-complete problem
- We know how to solve LP in polynomial time
- We will see that LP can be used to get approximate solutions to some NP-complete problems.

Weighted Vertex Cover

**Input:** Graph $G=(V,E)$ with non-negative weights $w_v$ on the vertices.

**Goal:** Find a minimum-cost set of vertices $S$, such that all the edges are covered. An edge is covered iff at least one of its endpoints is in $S$.

**Recall:** Weighted Vertex Cover is NP-complete. The best known approximation factor is $2 - 1/\sqrt{\log|V|}$.

Weighted Vertex Cover

**Variables:** for each $v \in V$, $x_v$ – is $v$ in the cover?

**Min** $\sum_{v \in V} w_v x_v$

**s.t.**

- $x_v + x_u \geq 1$, $\forall (u,v) \in E$
- $x_v \in \{0,1\}$ $\forall v \in V$

The LP Relaxation

This is not a linear program: the constraints of type $x_v \in \{0,1\}$ are not linear. We got an LP with integrality constraints on variables – an integer linear program (IP) that is NP-hard to solve.

However, if we replace the constraints $x_v \in \{0,1\}$ by $x_v \geq 0$ and $x_v \leq 1$, we will get a linear program.

The resulting LP is called a Linear Relaxation of the IP, since we relax the integrality constraints.
LP Relaxation of Weighted Vertex Cover

\[
\min \sum_{v \in V} w_v x_v \\
\text{s.t.} \\
x_v + x_u \geq 1, \quad \forall (u,v) \in E \\
x_v \geq 0, \quad \forall v \in V \\
x_v \leq 1, \quad \forall v \in V
\]

LP Relaxation of Weighted Vertex Cover - example

Consider the case of a 3-cycle in which all weights are 1.

An optimal VC has cost 2 (any two vertices)
An optimal relaxation has cost 3/2 (for all three vertices \(x_v = 1/2\))
The LP and the IP are different problems. Can we still learn something about Integral VC?

Why LP Relaxation Is Useful?

The optimal value of LP-solution provides a bound on the optimal value of the original optimization problem. \(\text{OPT}_{\text{LP}}\) is always better than \(\text{OPT}_{\text{IP}}\) (why?)

Therefore, if we find an integral solution within a factor \(r\) of \(\text{OPT}_{\text{LP}}\), it is also an \(r\)-approximation of the original problem.

It can be done by ‘wise’ rounding.

Approximation of Weighted Vertex Cover Using LP-Rounding

1. Solve the LP-Relaxation.
2. Let \(S\) be the set of all the vertices \(v\) with \(x_v \geq 1/2\). Output \(S\) as the solution.

Analysis: The solution is feasible: for each edge \(e = (u,v)\), either \(x_u \geq 1/2\) or \(x_v \geq 1/2\)

The value of the solution is:

\[
\sum_{v \in S} w_v = \sum_{v | x_v \geq 1/2} w_v \\
\leq 2 \sum_{v \in V} w_v x_v = 2 \cdot \text{OPT}_{\text{LP}}
\]

Since \(\text{OPT}_{\text{LP}} \leq \text{OPT}_{\text{VC}}\), the cost of the solution is \(\leq 2 \cdot \text{OPT}_{\text{VC}}\).
Linear Programming -Summary

- Of great practical importance to solve linear programs:
  - they model important practical problems
    - production, approximating the solution of inconsistent equations, manufacturing, network design, flow control, resource allocation.
  - solving an LP is often an important component of solving or approximating the solution to an integer linear programming problem.
- Can be solved in poly-time, but the simplex algorithm works very well in practice.
- One problem where you really do not want to roll your own code.