29 Linear Programming

- Many problems take the form of optimizing an objective, given limited resources and competing constraints.
- If we can specify
  - the objective as a linear function of certain variables, and
  - the constraints on resources as equalities or inequalities on those variables,

then we have a linear-programming problem.

A political problem

- You are a politician trying to win an election.
- Your district has three different types of areas — urban, suburban, and rural.
- These areas have, respectively, 100,000, 200,000, and 50,000 registered voters.
- You would like at least half the registered voters in each of the three regions to vote for you.
- Certain issues may be more effective in winning votes in certain places.
• The primary issues are building more roads, gun control, farm subsidies, and a gasoline tax dedicated to improved public transit

• According to your campaign staff’s research, you can estimate how many votes you win or lose from each population segment by spending $1,000 on advertising on each issue

• Each table entry indicates the number of thousands of voters who would be won over by spending $1,000 on advertising in support of a particular issue

<table>
<thead>
<tr>
<th>policy</th>
<th>urban</th>
<th>suburban</th>
<th>rural</th>
</tr>
</thead>
<tbody>
<tr>
<td>build roads</td>
<td>−2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>gun control</td>
<td>8</td>
<td>2</td>
<td>−5</td>
</tr>
<tr>
<td>farm subsidies</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>gasoline tax</td>
<td>10</td>
<td>0</td>
<td>−2</td>
</tr>
</tbody>
</table>

• Negative entries denote votes that would be lost

• Your task is to figure out the minimum amount of money that you need to spend in order to win 50,000 urban votes, 100,000 suburban votes, and 25,000 rural votes
You could, e.g., devote $20,000 of advertising to building roads, $0 to gun control, $4,000 to farm subsidies, and $9,000 to a gasoline tax.

You would win \(20(-2) + 0(8) + 4(0) + 9(10) = 50 \cdot 1000\) urban, \(20(5) + 0(2) + 4(0) + 9(0) = 100 \cdot 1000\) suburban, and \(20(3) + 0(-5) + 4(10) + 9(-2) = 82 \cdot 1000\) rural votes.

You would win the exact number of votes desired in the urban and suburban areas and more than enough votes in the rural area.

In order to garner these votes, we have paid \(20 + 0 + 4 + 9 = 33 \cdot $1000\) for advertising.

In order to develop a systematic method of optimizing the cost/benefit, we formulate this question mathematically.

We introduce 4 variables:
- \(x_1\) is the number of thousands of dollars spent on advertising on building roads,
- \(x_2\) is the number of thousands of dollars spent on advertising on gun control,
- \(x_3\) is the number of thousands of dollars spent on advertising on farm subsidies, and
- \(x_4\) is the number of thousands of dollars spent on advertising on a gasoline tax.
• We can write the requirement that we win at least 50,000 urban votes as
\[-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50\]
• Similarly, we can write the requirements that we win at least 100,000 suburban votes and 25,000 rural votes as
\[5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100\]
and
\[3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25\]
• Any setting of the variables \(x_1, x_2, x_3, x_4\) that satisfies these inequalities yields a strategy that wins a sufficient number of each type of vote.

• In order to keep costs as small as possible, you would like to minimize the amount spent on advertising
• That is, you want to minimize the expression
\[x_1 + x_2 + x_3 + x_4\]
• Although negative advertising often occurs in political campaigns, there is no such thing as negative-cost advertising
• Consequently, we require that
\[x_1 \geq 0, x_2 \geq 0, x_3 \geq 0\] and \(x_4 \geq 0\)
Combining the inequalities with the objective of minimizing total cost, we obtain what is known as a “linear program.”

We format this problem as

\[
\begin{align*}
\text{minimize} & \quad x_1 + x_2 + x_3 + x_4 \\
\text{subject to} & \quad -2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50 \\
& \quad 5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100 \\
& \quad 3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25 \\
& \quad x_1, x_2, x_3, x_4 \geq 0
\end{align*}
\]

The solution of this linear program yields your optimal strategy.

General linear programs

In the general linear-programming problem, we wish to optimize a linear function subject to a set of linear inequalities.

Given a set of real numbers \(a_1, a_2, \ldots, a_n\) and a set of variables \(x_1, x_2, \ldots, x_n\), we define a linear function \(f\) on those variables by

\[
f(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n = \sum_{j=1}^{n} a_j x_j
\]
If $b$ is a real number and $f$ is a linear function, then the equation
\[ f(x_1, x_2, \ldots, x_n) = b \]
is a **linear equality** and
\[ f(x_1, x_2, \ldots, x_n) \leq b \quad \text{and} \quad f(x_1, x_2, \ldots, x_n) \geq b \]
are **linear inequalities**.

- The term **linear constraints** denotes both
- Linear programming doesn’t allow strict inequalities
- Formally, a linear-programming problem is the problem of optimizing a linear function subject to a finite set of linear constraints

---

**An overview of linear programming**

- In order to describe properties of and algorithms for linear programs (LP), we express them in canonical forms
- We shall use two forms, **standard** and **slack**
- Informally, a LP in standard form is the maximization of a linear function subject to linear **inequalities**
- whereas a LP in slack form is the maximization of a linear function subject to linear **equalities**
Let us first consider the following linear program with two variables:

maximize \( x_1 + x_2 \)

subject to

\[
\begin{align*}
4x_1 - x_2 &\leq 8 \\
2x_1 + x_2 &\leq 10 \\
5x_1 - 2x_2 &\geq -2 \\
x_1, x_2 &\geq 0
\end{align*}
\]

We call any setting of the variables \( x_1 \) and \( x_2 \) that satisfies all the constraints a **feasible solution** to the linear program.

If we graph the constraints in the \((x_1, x_2)\)-Cartesian coordinate system, we see that the set of feasible solutions forms a convex region in the two-dimensional space.

An intuitive definition of a convex region is that it fulfills the requirement that for any two points in the region, all points on a line segment between them are also in the region.

We call this convex region the **feasible region** and the function we wish to maximize the **objective function**.
We could evaluate the objective function \( x_1 + x_2 \) at each point in the feasible region; we call the value of the objective function at a particular point the **objective value**. Then identify a point that has the maximum objective value as an optimal solution. For this example (and for most LPs), the feasible region contains an infinite number of points. We need to determine an efficient way to find a point that achieves the maximum objective value without explicitly evaluating the objective at every point in the feasible region.
Because the feasible region is bounded, there must be some max value \( z \) for which the intersection of the line \( x_1 + x_2 = z \) and the feasible region is nonempty.

Any point at which this occurs is an optimal solution to the LP, which in this case is the point \( x_1 = 2 \) and \( x_2 = 6 \) with objective value 8.

It is no accident that an optimal solution to the LP occurs at a vertex of the feasible region.

The maximum value of \( z \) for which the line \( x_1 + x_2 = z \) intersects the feasible region must be on the boundary of the feasible region.

Thus the intersection of this line with the boundary of the feasible region is either a single vertex or a line segment.

If the intersection is a single vertex, then there is just one optimal solution, and it is that vertex.

If the intersection is a line segment, every point on that line segment must have the same objective value.

In particular, both endpoints of the line segment are optimal solutions.

Each endpoint of a line segment is a vertex, so there is an optimal solution at a vertex also now.
29.1 Standard and slack forms

- In standard form, we are given \( n \) real numbers \( c_1, c_2, \ldots, c_n \), \( m \) real numbers \( b_1, b_2, \ldots, b_m \) and \( mn \)
real numbers \( a_{ij} \) for \( i = 1, 2, \ldots, m \) and \( j = 1, 2, \ldots, n \).
- We wish to find \( n \) real numbers \( x_1, x_2, \ldots, x_n \) that maximize \( \sum_{j=1}^{n} c_j x_j \)
subject to \( \sum_{j=1}^{n} a_{ij} x_j \leq b_i \) for \( i = 1, 2, \ldots, m \)
\( x_j \geq 0, \quad j = 1, 2, \ldots, n \)

- An arbitrary LP need not have non-negativity constraints, but standard form requires them.
- If we create
  - an \( m \times n \) matrix \( A = (a_{ij}) \),
  - an \( m \) -vector \( b = (b_i) \),
  - an \( n \) -vector \( c = (c_j) \), and
  - an \( n \) -vector \( x = (x_j) \),
then we can rewrite the standard form LP as
maximize \( c^T x \)
subject to \( Ax \leq b \)
\( x \geq 0 \)
Converting linear programs into standard form

- We can always convert a LP into standard form
  1. The objective function might be a minimization rather than a maximization
  2. There might be variables without nonnegativity constraints
  3. There might be equality constraints, which have an equal sign rather than a ≤ sign
  4. There might be inequality constraints, but instead of having ≤ sign, they have a ≥ sign

When converting one LP $L$ into another LP $L'$, we would like the property that an optimal solution to $L'$ yields an optimal solution to $L$

To capture this idea, we say that two maximization LPs $L$ and $L'$ are equivalent if
- for each feasible solution $\bar{x}$ to $L$ with objective value $z$, there is a corresponding feasible solution $\bar{x}'$ to $L'$ with objective value $z$, and
- for each feasible solution $\bar{x}'$ to $L'$ with objective value $z$, there is a corresponding feasible solution $\bar{x}$ to $L$ with objective value $z$
To convert a minimization LP $L$ into an equivalent maximization LP $L'$, we simply negate the coefficients in the objective function.

Since $L$ and $L'$ have identical sets of feasible solutions and, for any feasible solution, the objective value in $L$ is the negative of that in $L'$, these two LPs are equivalent.

E.g., if we have the LP

\[
\text{minimize } -2x_1 + 3x_2 \\
\text{subject to } x_1 + x_2 = 7 \\
\quad \quad \quad \quad x_1 - 2x_2 \leq 4 \\
\quad \quad \quad \quad x_1 \geq 0
\]

Negating the coefficients of the objective function, we obtain

\[
\text{maximize } 2x_1 - 3x_2 \\
\text{subject to } x_1 + x_2 = 7 \\
\quad \quad \quad \quad x_1 - 2x_2 \leq 4 \\
\quad \quad \quad \quad x_1 \geq 0
\]

Suppose that some variable $x_j$ does not have a nonnegativity constraint.
• Then, we replace each occurrence of \( x_j \) by \( x_j' - x_j'' \), and add the nonnegativity constraints \( x_j' \geq 0 \) and \( x_j'' \geq 0 \).

• Thus, if the objective function has a term \( c_j x_j \), we replace it by \( c_j x_j' - c_j x_j'' \).

• If constraint \( i \) has a term \( a_{ij} x_j \), we replace it by \( a_{ij} x_j' - a_{ij} x_j'' \).

• Any feasible solution \( \bar{x} \) to the new LP corresponds to a feasible solution \( \bar{x} \) to the original LP with \( \bar{x}_j = \bar{x}_j' - \bar{x}_j'' \) and with the same objective value.

• Also, any feasible solution \( \bar{x} \) to the original LP corresponds to one \( \bar{x} \) to the new LP
  – with \( \bar{x}_j' = \bar{x}_j \) and \( \bar{x}_j'' = 0 \) if \( \bar{x}_j \geq 0 \), or
  – with \( \bar{x}_j' = \bar{x}_j \) and \( \bar{x}_j' = 0 \) if \( \bar{x}_j < 0 \).

• The two LPs have the same objective value regardless of the sign of \( \bar{x}_j \)

• Thus, the two LPs are equivalent

• Apply this conversion scheme to each variable that does not have a nonnegativity constraint to yield an equivalent LP in which all variables have nonnegativity constraints.
Continuing the example, we want to ensure that each variable has a corresponding nonnegativity constraint.

- Variable $x_1$ has such a constraint, but $x_2$ doesn’t.
- Therefore, we replace $x_2$ by two variables $x'_2$ and $x''_2$, and we modify the LP to obtain

$$\begin{align*}
\text{maximize} & \quad 2x_1 - 3x'_2 + 3x''_2 \\
\text{subject to} & \quad x_1 + x'_2 - x''_2 = 7 \\
& \quad x_1 - 2x'_2 + 2x''_2 \leq 4 \\
& \quad x_1, x'_2, x''_2 \geq 0
\end{align*}$$

Let us convert equality constraints into inequality constraints.

- Suppose that a LP has an equality constraint $f(x_1, x_2, \ldots, x_n) = b$.
- Since $x = y$ if and only if both $x \geq y$ and $x \leq y$, we can replace this equality constraint by the pair of inequality constraints $f(x_1, x_2, \ldots, x_n) \geq b$ and $f(x_1, x_2, \ldots, x_n) \leq b$.
- Repeating this conversion for each equality constraint yields a linear program in which all constraints are inequalities.
Finally, we can convert the \( \geq \) constraints to \( \leq \) constraints by multiplying these constraints through by \(-1\).

I.e., any inequality of the form
\[
\sum_{j=1}^{n} a_{ij}x_j \geq b_i
\]
is equivalent to
\[
\sum_{j=1}^{n} -a_{ij}x_j \leq -b_i
\]

Thus, we obtain an equivalent \( \leq \) constraint.

Finishing our example, we replace the equality constraint by two inequalities, obtaining

\[
\begin{align*}
\text{maximize} & \quad 2x_1 - 3x_2' + 3x_2'' \\
\text{subject to} & \quad x_1 + x_2' - x_2'' \leq 7 \\
& \quad x_1 + x_2' - x_2'' \geq 7 \\
& \quad x_1 - 2x_2' + 2x_2'' \leq 4 \\
& \quad x_1, x_2', x_2'' \geq 0
\end{align*}
\]
Finally, we negate the $\geq$ constraint

For consistency in variable names, we rename $x_2'$ to $x_2$ and $x_2''$ to $x_3$, obtaining the standard form

maximize $2x_1 - 3x_2 + 3x_3$
subject to $x_1 + x_2 - x_3 \leq 7$
$-x_1 - x_2 + x_3 \leq -7$
$x_1 - 2x_2 + 2x_3 \leq 4$
$x_1, x_2, x_3 \geq 0$

Converting linear programs into slack form

To efficiently solve a LP with the simplex algorithm, we express it in a form in which some of the constraints are equality constraints

We convert it into a form in which the non-negativity constraints are the only inequality constraints

The remaining constraints are equalities

Consider an inequality constraint

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i$$
• We introduce a new variable \( s \) and rewrite the inequality as the two constraints
\[
s = b_i - \sum_{j=1}^{n} a_{ij} x_j \\
\quad s \geq 0
\]

• We call \( s \) a **slack variable** because it measures the **slack**, or difference, between the left-hand and right-hand sides of the original inequality.

• The original inequality is true if and only if both the new equation and the new inequality are true:
  – We can obtain an equivalent LP in which the only inequality constraints are the nonnegativity constraints.
  – Converting from standard to slack form, we let \( x_{n+1} \) denote the slack variable associated with the \( i \)th inequality.
• The \( i \)th constraint is therefore
\[
x_{n+1} = b_i - \sum_{j=1}^{n} a_{ij} x_j
\]
along with the nonnegativity constraint \( x_{n+1} \geq 0 \).
E.g., for the LP described above, we introduce slack variables $x_4, x_5,$ and $x_6$, obtaining

maximize $2x_1 - 3x_2 + 3x_3$

subject to

\begin{align*}
x_4 &= 7 - x_1 - x_2 + x_3 \\
x_5 &= -7 + x_1 + x_2 - x_3 \\
x_6 &= 4 - x_1 + 2x_2 - 2x_3 \\
x_1, x_2, x_3, x_4, x_5, x_6 &\geq 0
\end{align*}

- The variables on the LHS of the equalities are \textit{basic variables} and those on the RHS \textit{nonbasic variables}

29.2 Formulating problems as linear programs

- Once we cast a problem as a polynomial-sized linear program, we can solve it in polynomial time by the ellipsoid algorithm or interior-point methods

- Several linear-programming software packages can solve problems efficiently, so that once the problem is in the form of a linear program, such a package can solve it
Shortest paths

- We formulate the single-pair shortest-path problem (extension to the general single-source shortest-paths problem is an exercise)
- We are given
  - a weighted, directed graph $G = (V, E)$, with weight function $w : E \rightarrow \mathbb{R}$ mapping edges to real-valued weights,
  - a source vertex $s$,
  - and destination vertex $t$

We wish to compute the value $d_t$, which is the weight of a shortest path from $s$ to $t$

To express this problem as a LP, we need to determine a set of variables and constraints that define when we have a shortest path from $s$ to $t$

Fortunately, the Bellman-Ford (BF) algorithm does exactly this

When the BF algorithm terminates, it has computed, for each vertex $v$, a value $d_v$ such that for each edge $(u, v) \in E$, we have

$$d_v \leq d_u + w(u, v)$$
• The source vertex initially receives a value \( d_s = 0 \), which never changes
• Thus we obtain the following LP to compute the shortest-path weight from \( s \) to \( t \):

\[
\begin{align*}
& \text{maximize} & d_t \\
& \text{subject to} & d_v \leq d_u + w(u, v) & \text{for each edge } (u, v) \in E \\
& & d_s = 0
\end{align*}
\]

This LP maximizes an objective function when it is supposed to compute shortest paths
• We do not want to minimize the objective, since then setting \( d_v = 0 \) for all \( v \in V \) would yield an optimal solution to the LP without solving the shortest-paths problem
• We maximize because an optimal solution to the shortest-paths problem sets each \( \tilde{d}_v \) to

\[
\min_{u: (u, v) \in E} \{ \tilde{d}_u + w(u, v) \},
\]

s.t. \( \tilde{d}_v \) is the largest value \( \leq \) all of the values in the set \( \{ \tilde{d}_u + w(u, v) \} \)
We want to maximize $d_v$ for all vertices $v$ on a shortest path from $s$ to $t$ subject to these constraints on all vertices $v$, and maximizing $d_t$ achieves this goal.

This linear program has $|V|$ variables $d_v$, one for each vertex $v \in V$.

It also has $|E| + 1$ constraints: one for each edge, plus the additional constraint that the source vertex’s shortest-path weight always has the value 0.

**Maximum flow**

We are given
- a directed graph $G = (V, E)$ in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$, and
- two distinguished vertices: a source $s$ and a sink $t$

A flow is a nonnegative real-valued function $f: V \times V \to \mathbb{R}$ that satisfies the capacity constraint and flow conservation.
A maximum flow is a flow that satisfies these constraints and maximizes the flow value
- the total flow coming out of the source minus the total flow into the source

A flow, therefore, satisfies linear constraints, and the value of a flow is a linear function.

Recall also that we assume that $c(u, v) = 0$ if $(u, v) \notin E$ and that there are no antiparallel edges.

We can express the maximum-flow problem as:

\[
\begin{align*}
\text{maximize} & \quad \sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} \\
\text{subject to} & \quad f_{uv} \leq c(u, v) \text{ for each } u, v \in V \\
& \quad \sum_{v \in V} f_{vu} = \sum_{v \in V} f_{uv} \text{ for each } u \in V - \{s, t\} \\
& \quad f_{uv} \geq 0 \text{ for each } u, v \in V
\end{align*}
\]

This LP has $|V|^2$ variables for the flow between each pair of vertices, and it has $2|V|^2 + |V| - 2$ constraints.
29.4 Duality

- Given a LP in which the objective is to maximize, we shall describe how to formulate a dual LP in which
  - the objective is to minimize and
  - whose optimal value is identical to that of the original LP
- When referring to dual LPs, we call the original linear program the primal

- Given a primal LP in standard form
  maximize $\sum_{j=1}^{n} c_j x_j$
  subject to $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$ for $i = 1, 2, \ldots, m$
  $x_j \geq 0$, $j = 1, 2, \ldots, n$

- we define the dual LP as
  minimize $\sum_{i=1}^{m} b_i y_i$
  subject to $\sum_{i=1}^{m} a_{ij} y_i \geq c_j$ for $j = 1, 2, \ldots, n$
  $y_i \geq 0$, $i = 1, 2, \ldots, m$
• To form the dual,
  – we change the max to a min,
  – exchange the roles of coefficients on the right-hand sides and the objective function,
  – and replace each $\leq$ by a $\geq$
• Each of the $m$ constraints in the primal has an associated variable $y_i$ in the dual, and each of the $n$ constraints in the dual has an associated variable $x_j$ in the primal

maximize $3x_1 + x_2 + 2x_3$
subject to $x_1 + x_2 + 3x_3 \leq 30$
         $2x_1 + 2x_2 + 5x_3 \leq 24$
         $4x_1 + x_2 + 2x_3 \leq 36$
         $x_1, x_2, x_3 \geq 0$
• The dual of this LP is
minimize $30y_1 + 24y_2 + 36y_3$
subject to $y_1 + 2y_2 + 4y_3 \geq 3$
         $y_1 + 2y_2 + y_3 \geq 1$
         $3y_1 + 5y_2 + 2y_3 \geq 2$
         $x_1, x_2, x_3 \geq 0$
Lemma 29.8 (Weak linear-programming duality)
Let $\bar{x}$ be any feasible solution to the primal LP and let $\bar{y}$ be any feasible solution to the dual LP. Then,
$$
\sum_{j=1}^{n} c_j \bar{x}_j \leq \sum_{i=1}^{m} b_i \bar{y}_i.
$$

Corollary 29.9
Let $\bar{x}$ be a feasible solution to a primal LP and let $\bar{y}$ be a feasible solution to the corresponding dual LP. If $\sum_{j=1}^{n} c_j \bar{x}_j = \sum_{i=1}^{m} b_i \bar{y}_i$ then $\bar{x}$ and $\bar{y}$ are optimal solutions to the primal and dual linear programs, respectively.

Proof By Lemma 29.8, the objective value of a feasible solution to the primal cannot exceed that of a feasible solution to the dual.

The primal LP is a maximization problem and the dual is a minimization problem. Thus, if feasible solutions $\bar{x}$ and $\bar{y}$ have the same objective value, neither can be improved.

- Theorem 29.10 states that the optimal value of the dual LP is always equal to the optimal value of the primal LP.