Algorithm Design, Analysis & Complexity

Lecture 7 - Linear Programming

Koushik Pal

University of Toronto

July 6, 2021

Linear Function and Constraint

Definition

Given a set of real numbers $c_1, \dots, c_n$ and a set of variables $x_1, \dots x_n$, we define a **linear function** f on those variables by

$$f(x_1, \dots, x_n) = c_1 x_1 + \cdots + c_n x_n = \sum_{j=1}^{n} c_j x_j .$$

If b is a real number and f is a linear function, then

$f(x_1, \dots, x_n) = b$ is called a **linear equality**, and

$f(x_1, \dots, x_n) \le b$ and $f(x_1, \dots, x_n) \ge b$ are called **linear inequalities**.

We use the general term **linear constraints** to denote either linear equalities or linear inequalities.

Linear Programming Problem (LPP)

Definition

A **linear programming problem (LPP)** is the problem of minimizing or maximizing a linear function subject to a finite set of linear constraints.

Correspondingly, the linear program is called a **minimization LPP** or a **maximization LPP**.

If all the coefficients in both the objective function and the constraints are integers and the variables are constrained to assume only integral values, the corresponding LPP is called an **Integer Programming Problem (IPP)**.

Feasible Solution

Definition

Any value $(\bar{x}_1, \ldots, \bar{x}_n)$ of the variables that satisfies all the constraints of an LPP is called a **feasible solution** of the LPP.

The set of all feasible solutions is a **convex region** and is called the **feasible region**.

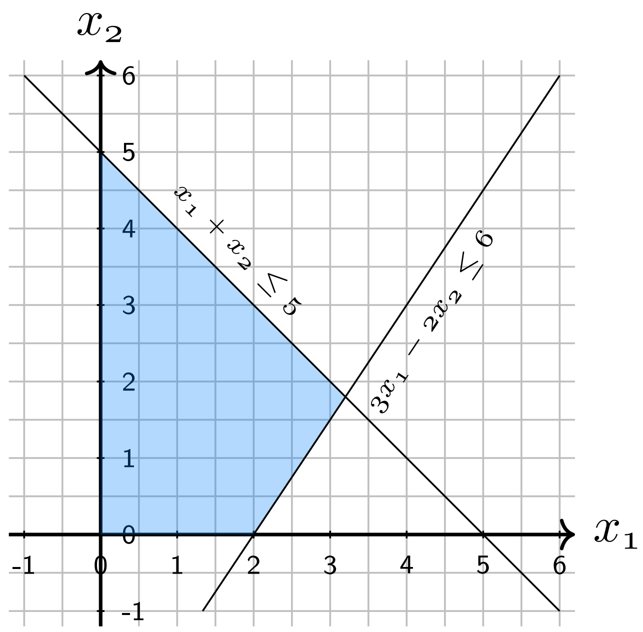
The function we wish to minimize or maximize is called the **objective function** and its value at any point $(\bar{x}_1, \ldots, \bar{x}_n)$ is called the **objective value** at that point.

A point that has the minimum (maximum) objective value for a minimization (maximization) LPP is called an **optimal solution**.

An LPP that has no feasible solution is called **infeasible**; otherwise it is called **feasible**. If an LPP has feasible solutions but no finite optimal objective value, it is called **unbounded**.

Example 1

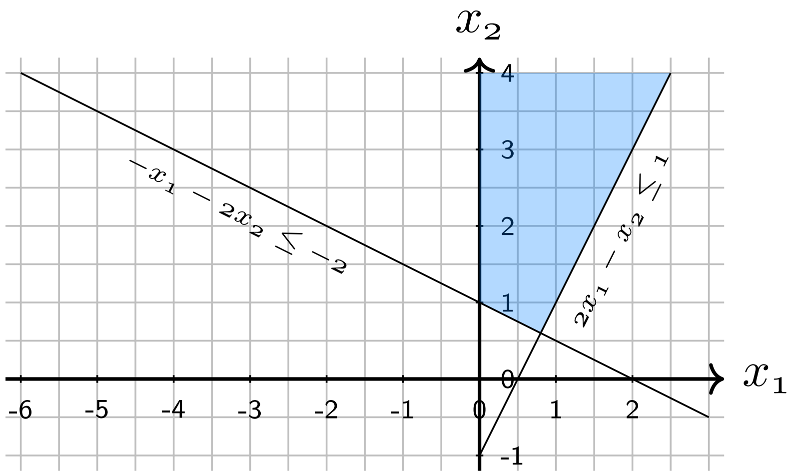
$$\begin{array}{rrcl} \text{maximize} & 2x_1 + 3x_2 & & \\ \text{subject to} & & & \\ & 3x_1 - 2x_2 & \leq & 6 \\ & x_1 + x_2 & \leq & 5 \\ & x_1, x_2 & \geq & 0. \end{array}$$



Bounded feasible region with optimal solution $(0, 5)$.

Example 2

$$\begin{array}{rrcr} \text{maximize} & x_1 + x_2 & & \\ \text{subject to} & & & \\ & 2x_1 - x_2 & \leq & 1 \\ & -x_1 - 2x_2 & \leq & -2 \\ & x_1, x_2 & \geq & 0. \end{array}$$



Unbounded feasible region with no optimal value.

LPP Representations - Standard Form

Standard Form

$$
\begin{array}{rl}
\text{maximize} & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} &
\end{array}
$$

$$
\begin{array}{rclll}
\sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for} & i = 1, \ldots, m \\
x_j & \geq & 0 & \text{for} & j = 1, \ldots, n.
\end{array}
$$

In matrix-vector notation,

$$
\begin{array}{rl}
\text{maximize} & \boldsymbol{c}^T \boldsymbol{x} \\
\text{subject to} &
\end{array}
$$

$$
\begin{array}{rcl}
A\boldsymbol{x} & \leq & \boldsymbol{b} \\
\boldsymbol{x} & \geq & \boldsymbol{0},
\end{array}
$$

where $\boldsymbol{c} = (c_1, \ldots, c_n)^T$, $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, $\boldsymbol{b} = (b_1, \ldots, b_m)^T$, and $A = (a_{ij})_{i=1,j=1}^{i=m,j=n}$.

Converting an LPP into Standard Form

Definition

Two LPPs L and L' are **equivalent** if for each feasible solution $\boldsymbol{x}$ to L with objective value z, there is a corresponding feasible solution $\boldsymbol{x}'$ to L' with objective value z, and vice versa.

Observation

1. *minimize* $\boldsymbol{c}^T\boldsymbol{x} \iff$ *maximize* $-\boldsymbol{c}^T\boldsymbol{x}$
2. $\sum_{j=1}^n a_{ij}x_j \geq b_i \iff \sum_{j=1}^n -a_{ij}x_j \leq -b_i$
3. $\sum_{j=1}^n a_{ij}x_j = b_i \iff \sum_{j=1}^n a_{ij}x_j \geq b_i$ *and* $\sum_{j=1}^n a_{ij}x_j \leq b_i$,
   $\iff \sum_{j=1}^n -a_{ij}x_j \leq -b_i$ *and* $\sum_{j=1}^n a_{ij}x_j \leq b_i$
4. x_i *is unbounded* $\iff x_i = x_i' - x_i''$ *with* $x_i', x_i'' \geq 0$.

LPP Representations - Slack Form

Definition

A **slack form** is a form where the only inequality constraints are the non-negativity constraints, rest all are equality constraints using slack variables.

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \iff s := b_i - \sum_{j=1}^{n} a_{ij}x_j \geq 0.$$

The new variable s is called a **slack variable** because it measures the slack between LHS and RHS.

When converting from a standard form to a slack form, denote the i^{th} slack variable by x_{n+i}. Thus, for $i = 1, \ldots, m$, we have

$$\begin{aligned} x_{n+i} &= b_i - \sum_{j=1}^{n} a_{ij}x_j \\ x_{n+i} &\geq 0. \end{aligned}$$

We call the variables on the left side of the equalities **basic variables**, and those on the right side **nonbasic variables**.

LPP Representations - Slack Form

We remove the words "maximize" and "subject to", and use the variable z to denote the value of the objective function.

We use N to denote the set of nonbasic variables and B to denote the basic variables. We always have that $|N| = n$, $|B| = m$ and $N \cup B = \{1, \ldots, n, \ldots, n+m\}$.

The equations in the slack form are indexed by the entries of B, the variables on the RHS are indexed by the entries of N, the variables b_i, c_j and a_{ij} denote the constant terms and coefficients, and we use the letter v to denote an optional constant term in the objective function.

LPP Representations - Slack Form

Then we can define a **slack form** by a tuple $(N, B, A, \boldsymbol{b}, \boldsymbol{c}, v)$, denoting the following LPP

$$
\begin{array}{rcll}
z & = & v + \sum_{j \in N} c_j x_j & \\
x_i & = & b_i - \sum_{j \in N} a_{ij} x_j & \text{for} \quad i \in B \\
x_k & \geq & 0 & \text{for} \quad k \in N \cup B.
\end{array}
$$

Note

Because we subtract the sum $\sum_{j \in N} a_{ij} x_j$, the values a_{ij} are actually the negatives of the coefficients as they "appear" in the slack form.

Example

$$\begin{aligned} z &= 5 - \frac{x_1}{7} - \frac{x_3}{8} + \frac{x_4}{10} \\ x_2 &= 7 + \frac{x_1}{8} + \frac{x_3}{7} - 2x_4 \\ x_5 &= 10 - \frac{x_1}{9} - \frac{2x_3}{3} + 3x_4 \\ x_1, x_2, x_3, x_4, x_5 &\geq 0. \end{aligned}$$

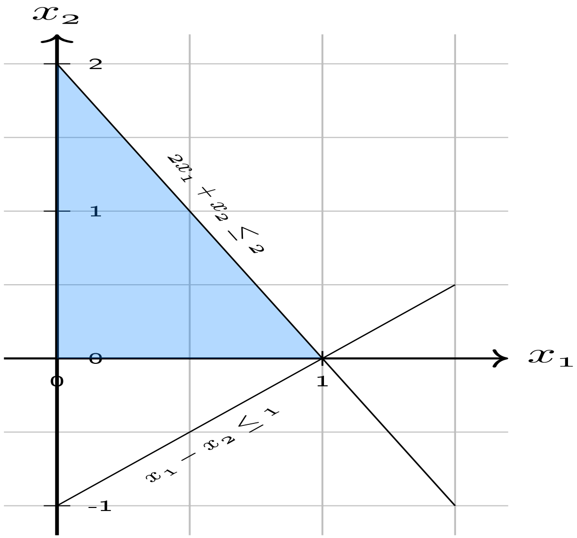
Here,

$$\begin{aligned} B &= \{2, 5\}, \quad N = \{1, 3, 4\}, \\ A &= \begin{pmatrix} a_{21} & a_{23} & a_{24} \\ a_{51} & a_{53} & a_{54} \end{pmatrix} = \begin{pmatrix} -\frac{1}{8} & -\frac{1}{7} & 2 \\ \frac{1}{9} & \frac{2}{3} & -3 \end{pmatrix}, \\ \boldsymbol{b} &= \begin{pmatrix} b_2 \\ b_5 \end{pmatrix} = \begin{pmatrix} 7 \\ 10 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} c_1 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} -\frac{1}{7} \\ -\frac{1}{8} \\ \frac{1}{10} \end{pmatrix}, \\ v &= 5. \end{aligned}$$

Simplex Algorithm

$$\begin{array}{ll} \text{maximize} & 5x_1 - 3x_2 \\ \text{subject to} & \end{array}$$

$$\begin{array}{rcl} x_1 - x_2 & \leq & 1 \\ 2x_1 + x_2 & \leq & 2 \\ x_1, x_2 & \geq & 0. \end{array}$$



Solution.

Step 0. Convert to slack form.

$$\begin{array}{rcl} z & = & 5x_1 - 3x_2 \\ x_3 & = & 1 - x_1 + x_2 \\ x_4 & = & 2 - 2x_1 - x_2 \\ x_1, x_2, x_3, x_4 & \geq & 0. \end{array}$$

Basic solution: $x_1 = x_2 = 0, x_3 = 1, x_4 = 2$. This is a feasible solution with objective value $z = 0$.

Simplex Algorithm

Step I. Select a nonbasic variable x_e whose coefficient in the objective function is positive, and increase the value of x_e as much as possible. This variable is called the entering variable.

In this example, $x_e = x_1$.

Step II. If we set x_1 to more than 1, then both x_3 and x_4 will become negative. So, the maximum we can increase x_1 by is 1. In this case, both the equations provide the same constraint on x_1. So, we can choose either. Otherwise, we pick the equation that imposes a stronger constraint. This basic variable on the left hand side of the chosen equation is called the leaving variable.

In this example, let us choose x_3 as the leaving variable.

Simplex Algorithm

Step III. Rewrite x_1 in terms of x_2 and x_3:

$$x_1 = 1 + x_2 - x_3.$$

Substitute this everywhere in the slack form to obtain an equivalent slack form.

$$\begin{aligned} z &= 5 + 2x_2 - 5x_3 \\ x_1 &= 1 + x_2 - x_3 \\ x_4 &= -3x_2 + 2x_3 \\ x_1, x_2, x_3, x_4 &\geq 0. \end{aligned}$$

Basic solution: $x_2 = x_3 = 0, x_1 = 1, x_4 = 0$. This is a feasible solution with objective value $z = 5$.

Step IV. Repeat Steps I, II and III as long as there is a variable with positive coefficient in the objective function.
New entering variable: x_2.
Tightest constraint is provided by Equation 2 since $\frac{0}{3} < \infty$.
So, x_4 is the leaving variable.

Simplex Algorithm

Write x_2 in terms of x_3 and x_4:

$$x_2 = \frac{2}{3}x_3 - \frac{1}{3}x_4.$$

Consequently, we obtain the following equivalent slack form

$$\begin{aligned}
z &= 5 - \frac{11}{3}x_3 - \frac{2}{3}x_4 \\
x_1 &= 1 - \frac{1}{3}x_3 - \frac{1}{3}x_4 \\
x_2 &= \frac{2}{3}x_3 - \frac{1}{3}x_4 \\
x_1, x_2, x_3, x_4 &\geq 0.
\end{aligned}$$

Basic solution: $x_3 = x_4 = 0, x_1 = 1, x_2 = 0$. This is a feasible solution with objective value $z = 5$.

Note that there are no more variables left with positive coefficient in the objective function. So, we stop.

Thus, Optimal solution = Basic solution = $(1, 0)$, and optimal objective value = $v = 5$.

Pivot

Before we present the general algorithm, we first define a subroutine called $\textsc{Pivot}$ that takes a slack form $(N, B, A, \boldsymbol{b}, \boldsymbol{c}, v)$, a leaving variable x_ℓ $(\ell \in B)$ and an entering variable x_e $(e \in N)$, and transforms the equations into an equivalent slack form $(\hat{N}, \hat{B}, \hat{A}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \hat{v})$ so that $\ell \in \hat{N}$ and $e \in \hat{B}$.

Pivot I

procedure $\textsc{Pivot}(N, B, A, \boldsymbol{b}, \boldsymbol{c}, v, \ell, e)$

  # Compute the coefficients of the equations for the new basic variable x_e

  Let $\hat{A}$ be a new $m \times n$ matrix

  $\hat{b}_e = \dfrac{b_\ell}{a_{\ell e}}$

  for $j \in N \setminus \{\ell\}$ **do**

    $\hat{a}_{ej} = \dfrac{a_{\ell j}}{a_{\ell e}}$

  $\hat{a}_{e\ell} = \dfrac{1}{a_{\ell e}}$

  # Compute the coefficients of the remaining constraints

  for $i \in B \setminus \{\ell\}$ **do**

    $\hat{b}_i = b_i - a_{ie}\hat{b}_e$

    for $j \in N \setminus \{\ell\}$ **do**

      $\hat{a}_{ij} = a_{ij} - a_{ie}\hat{a}_{ej}$

    $\hat{a}_{i\ell} = -a_{ie}\hat{a}_{e\ell}$

Pivot II

\# Compute the objective function

$\hat{v} = v + c_e \hat{b}_e$

for $j \in N \setminus \{\ell\}$ **do**

  $\hat{c}_j = c_j - c_e \hat{a}_{ej}$

$\hat{c}_\ell = -c_e \hat{a}_{e\ell}$

\# Compute new sets of basic and nonbasic variables

$\hat{N} = N \cup \{\ell\} \setminus \{e\}$

$\hat{B} = B \cup \{e\} \setminus \{\ell\}$

return $(\hat{N}, \hat{B}, \hat{A}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \hat{v})$

Simplex

Now, we present the actual Simplex algorithm in its full generality. The subroutine Initialize-Simplex used in the algorithm checks if the original problem is feasible. If not, it returns “infeasible” and halts. If yes, it returns an equivalent slack form for which the basic solution is feasible.

Simplex I

procedure Simplex$(A, \boldsymbol{b}, \boldsymbol{c})$
    $(N, B, A, \boldsymbol{b}, \boldsymbol{c}, v) = $ Initialize-Simplex $(A, \boldsymbol{b}, \boldsymbol{c})$
    Let Δ be a new vector of length m
    while some index $j \in N$ has $c_j > 0$ **do**
        Choose an index $e \in N$ for which $c_e > 0$
        for each index $i \in B$ **do**
            if $a_{ie} > 0$ **then**
                $\Delta_i = \dfrac{b_i}{a_{ie}}$
            else
                $\Delta_i = \infty$
        Choose an index $\ell \in B$ that minimizes Δ_ℓ
        if $\Delta_\ell == \infty$ **then**
            return "unbounded"
        else
            $(N, B, A, \boldsymbol{b}, \boldsymbol{c}, v) = $ Pivot $(N, B, A, \boldsymbol{b}, \boldsymbol{c}, v, \ell, e)$

Simplex II

for $i = 1 \ldots n$ **do**
    if $i \in B$ **then**
        $\bar{x}_i = b_i$
    else
        $\bar{x}_i = 0$
return $(\bar{x}_1, \ldots, \bar{x}_n)$