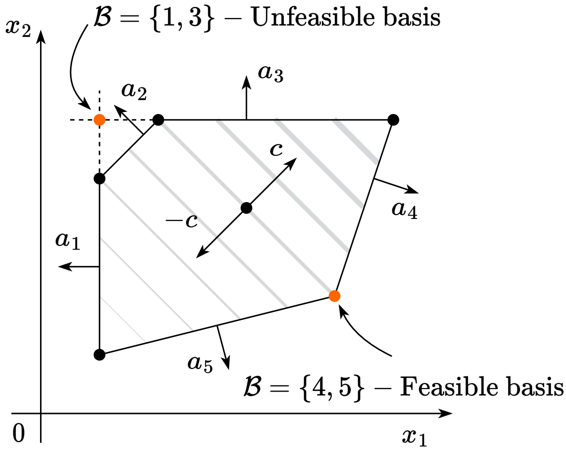
Two-Phase Simplex Method. Duality in LP

Daniil Merkulov

Optimization methods. MIPT

Simplex method

Geometry of simplex method

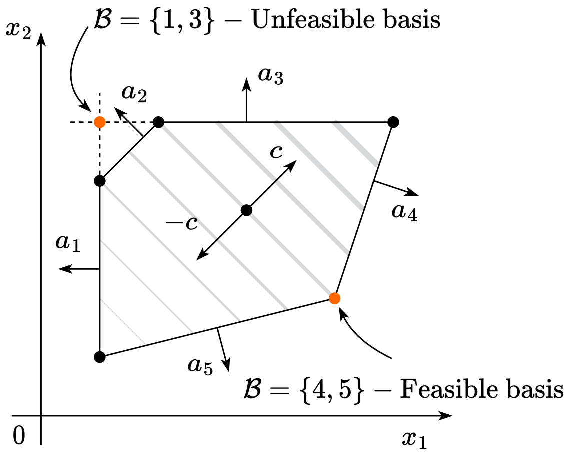


We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } &Ax \leq b
\end{aligned}
\tag{LP.Inequality}
$$

- Definition: a **basis** $\mathcal{B}$ is a subset of n (integer) numbers between 1 and m, so that $\text{rank} A_{\mathcal{B}} = n$.

Geometry of simplex method

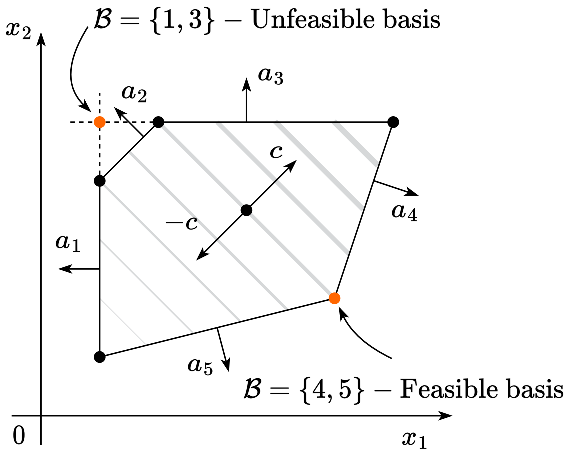


We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
&\text{s.t. } Ax \leq b
\end{aligned}
\tag{LP.Inequality}
$$

- Definition: a **basis** $\mathcal{B}$ is a subset of n (integer) numbers between 1 and m, so that $\text{rank} A_{\mathcal{B}} = n$.
- Note, that we can associate submatrix $A_{\mathcal{B}}$ and corresponding right-hand side $b_{\mathcal{B}}$ with the basis $\mathcal{B}$.

Geometry of simplex method

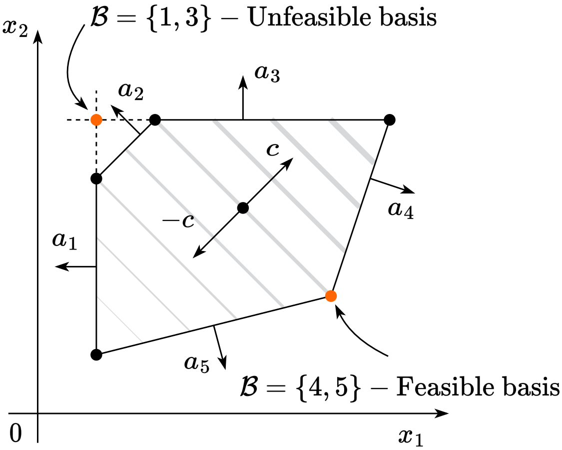


We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } & Ax \leq b
\end{aligned}
\tag{LP.Inequality}
$$

- Definition: a **basis** $\mathcal{B}$ is a subset of n (integer) numbers between 1 and m, so that $\text{rank} A_{\mathcal{B}} = n$.
- Note, that we can associate submatrix $A_{\mathcal{B}}$ and corresponding right-hand side $b_{\mathcal{B}}$ with the basis $\mathcal{B}$.
- Also, we can derive a point of intersection of all these hyperplanes from the basis: $x_{\mathcal{B}} = A^{-1}_{\mathcal{B}} b_{\mathcal{B}}$.

Geometry of simplex method

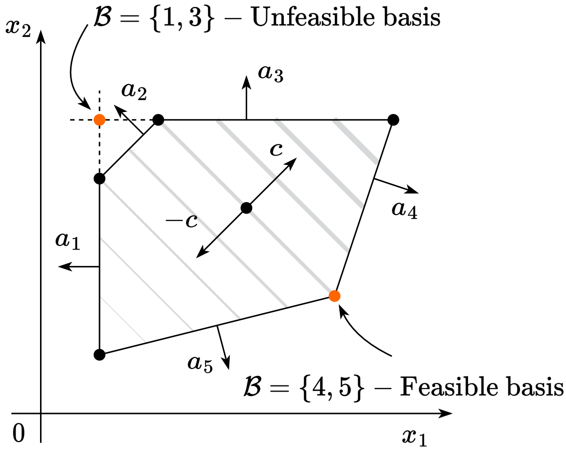


We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } &Ax \leq b
\end{aligned}
\tag{LP.Inequality}
$$

- Definition: a **basis** $\mathcal{B}$ is a subset of n (integer) numbers between 1 and m, so that $\text{rank} A_{\mathcal{B}} = n$.
- Note, that we can associate submatrix $A_{\mathcal{B}}$ and corresponding right-hand side $b_{\mathcal{B}}$ with the basis $\mathcal{B}$.
- Also, we can derive a point of intersection of all these hyperplanes from the basis: $x_{\mathcal{B}} = A^{-1}_{\mathcal{B}} b_{\mathcal{B}}$.
- If $Ax_{\mathcal{B}} \leq b$, then basis $\mathcal{B}$ is **feasible**.

Geometry of simplex method

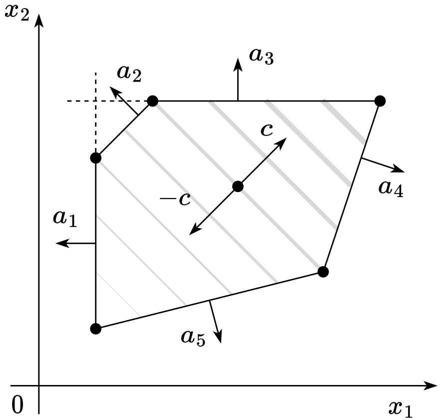


We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } & Ax \leq b
\end{aligned}
\tag{LP.Inequality}
$$

- Definition: a **basis** $\mathcal{B}$ is a subset of n (integer) numbers between 1 and m, so that $\text{rank} A_{\mathcal{B}} = n$.
- Note, that we can associate submatrix $A_{\mathcal{B}}$ and corresponding right-hand side $b_{\mathcal{B}}$ with the basis $\mathcal{B}$.
- Also, we can derive a point of intersection of all these hyperplanes from the basis: $x_{\mathcal{B}} = A^{-1}_{\mathcal{B}} b_{\mathcal{B}}$.
- If $Ax_{\mathcal{B}} \leq b$, then basis $\mathcal{B}$ is **feasible**.
- A basis $\mathcal{B}$ is optimal if $x_{\mathcal{B}}$ is an optimum of the LP.Inequality.

The solution of LP if exists lies in the corner

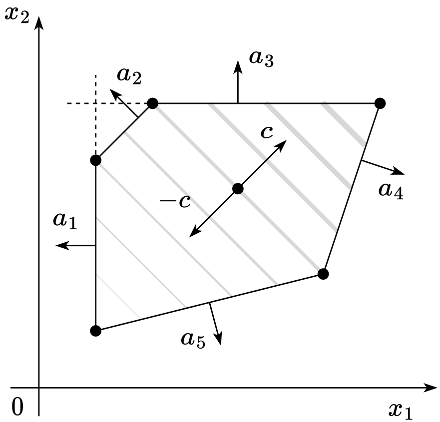


Theorem

1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point

The high-level idea of the simplex method is following:

The solution of LP if exists lies in the corner

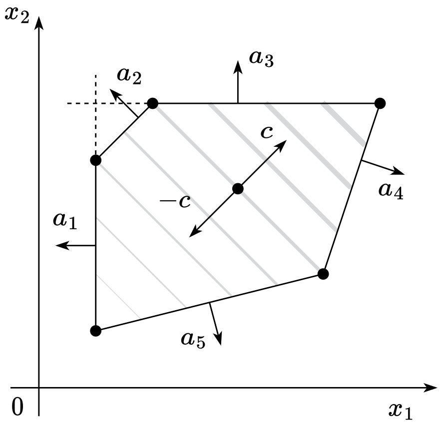


Theorem

1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
2. If Standard LP has solutions, then at least one such solution is a basic optimal point.

The high-level idea of the simplex method is following:

The solution of LP if exists lies in the corner

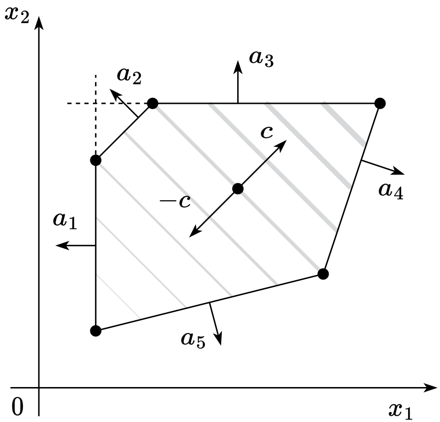


Theorem

1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
3. If Standard LP is feasible and bounded, then it has an optimal solution.

The high-level idea of the simplex method is following:

The solution of LP if exists lies in the corner

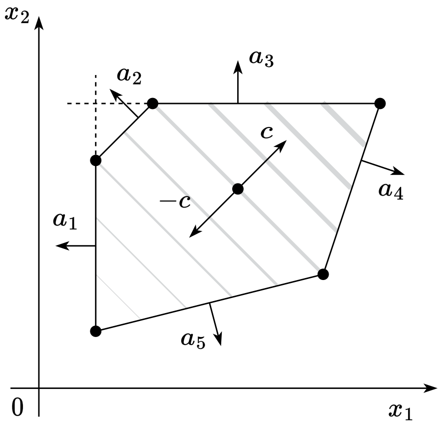


Theorem

1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
3. If Standard LP is feasible and bounded, then it has an optimal solution.

The high-level idea of the simplex method is following:

The solution of LP if exists lies in the corner



Theorem

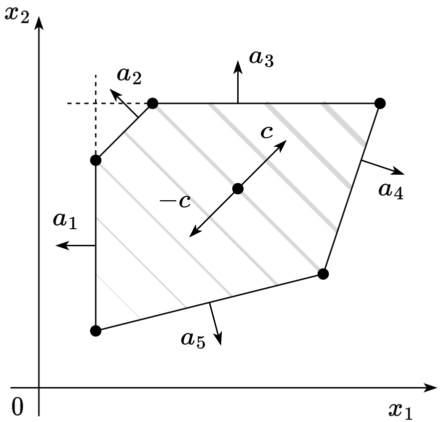
1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
3. If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

The high-level idea of the simplex method is following:

- Ensure, that you are in the corner.

The solution of LP if exists lies in the corner



Theorem

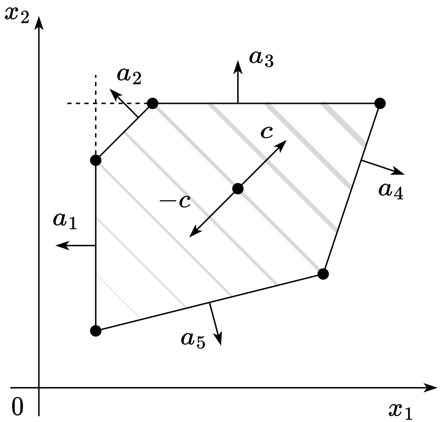
1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
3. If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

The high-level idea of the simplex method is following:

- Ensure, that you are in the corner.
- Check optimality.

The solution of LP if exists lies in the corner



Theorem

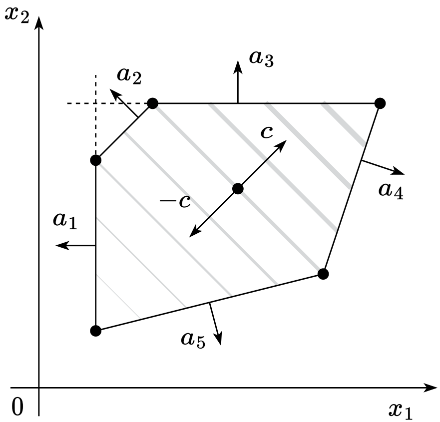
1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
3. If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

The high-level idea of the simplex method is following:

- Ensure, that you are in the corner.
- Check optimality.
- If necessary, switch the corner (change the basis).

The solution of LP if exists lies in the corner



Theorem

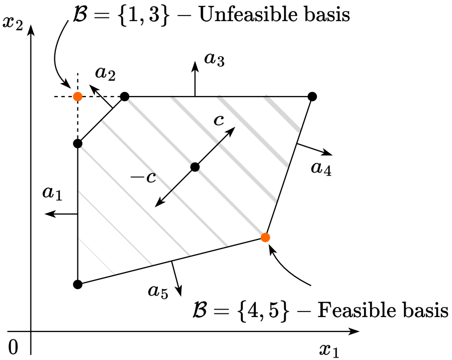
1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
3. If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

The high-level idea of the simplex method is following:

- Ensure, that you are in the corner.
- Check optimality.
- If necessary, switch the corner (change the basis).
- Repeat until converge.

Optimal basis



Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

$$
\lambda^T_{\mathcal{B}} A_{\mathcal{B}} = c^T \leftrightarrow \lambda^T_{\mathcal{B}} = c^T A_{\mathcal{B}}^{-1}
$$

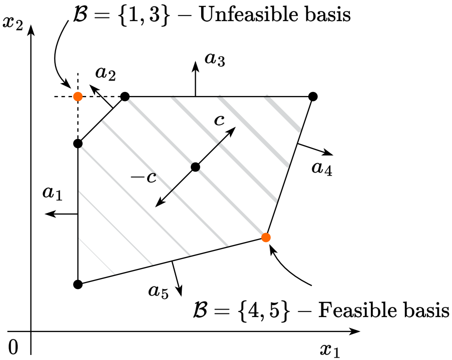
Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and $\mathcal{B}$ is feasible, then $\mathcal{B}$ is optimal.

Proof

$$
\exists x^* : Ax^* \leq b, c^T x^* < c^T x_{\mathcal{B}}
$$

Optimal basis



Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

$$
\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}
$$

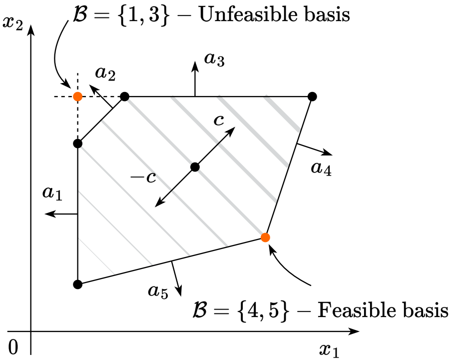
Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and $\mathcal{B}$ is feasible, then $\mathcal{B}$ is optimal.

Proof

$$
\exists x^* : Ax^* \leq b, c^T x^* < c^T x_{\mathcal{B}}
$$
$$
A_{\mathcal{B}} x^* \leq b_{\mathcal{B}}
$$

Optimal basis



Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_\mathcal{B}$:

$$\lambda^T_\mathcal{B} A_\mathcal{B} = c^T \leftrightarrow \lambda^T_\mathcal{B} = c^T A_\mathcal{B}^{-1}$$

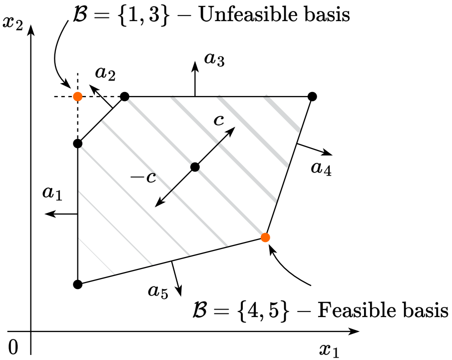
Theorem

If all components of $\lambda_\mathcal{B}$ are non-positive and $\mathcal{B}$ is feasible, then $\mathcal{B}$ is optimal.

Proof

$$\exists x^* : Ax^* \leq b, c^T x^* < c^T x_\mathcal{B}$$
$$A_\mathcal{B} x^* \leq b_\mathcal{B}$$
$$\lambda^T_\mathcal{B} A_\mathcal{B} x^* \geq \lambda^T_\mathcal{B} b_\mathcal{B}$$

Optimal basis



Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_\mathcal{B}$:

$$\lambda_\mathcal{B}^T A_\mathcal{B} = c^T \leftrightarrow \lambda_\mathcal{B}^T = c^T A_\mathcal{B}^{-1}$$

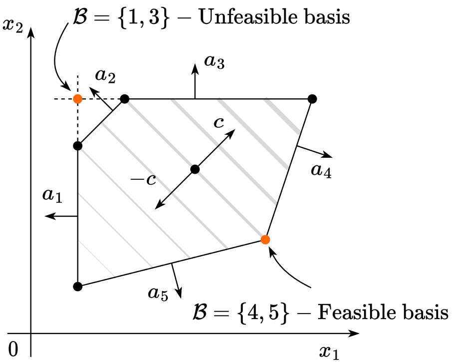
Theorem

If all components of $\lambda_\mathcal{B}$ are non-positive and $\mathcal{B}$ is feasible, then $\mathcal{B}$ is optimal.

Proof

$$\begin{aligned}
\exists x^* : Ax^* \leq b, c^T x^* < c^T x_\mathcal{B} \\
A_\mathcal{B} x^* \leq b_\mathcal{B} \\
\lambda_\mathcal{B}^T A_\mathcal{B} x^* \geq \lambda_\mathcal{B}^T b_\mathcal{B} \\
c^T x^* \geq \lambda_\mathcal{B}^T A_\mathcal{B} x_\mathcal{B}
\end{aligned}$$

Optimal basis



Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

$$\lambda^T_{\mathcal{B}} A_{\mathcal{B}} = c^T \leftrightarrow \lambda^T_{\mathcal{B}} = c^T A^{-1}_{\mathcal{B}}$$

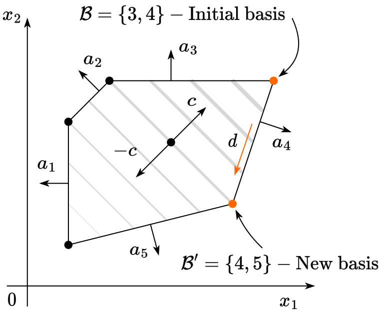
Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and $\mathcal{B}$ is feasible, then $\mathcal{B}$ is optimal.

Proof

$$\begin{aligned} \exists x^* : Ax^* \leq b, c^T x^* &< c^T x_{\mathcal{B}} \\ A_{\mathcal{B}} x^* &\leq b_{\mathcal{B}} \\ \lambda^T_{\mathcal{B}} A_{\mathcal{B}} x^* &\geq \lambda^T_{\mathcal{B}} b_{\mathcal{B}} \\ c^T x^* &\geq \lambda^T_{\mathcal{B}} A_{\mathcal{B}} x_{\mathcal{B}} \\ c^T x^* &\geq c^T x_{\mathcal{B}} \end{aligned}$$

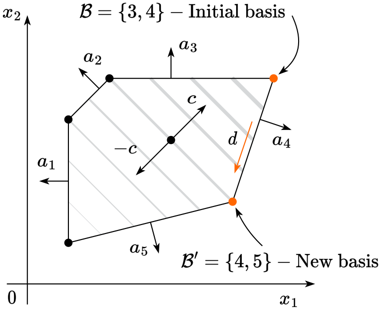
Changing basis



Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$

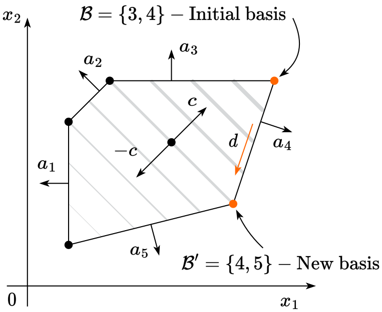
Changing basis



Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

Changing basis

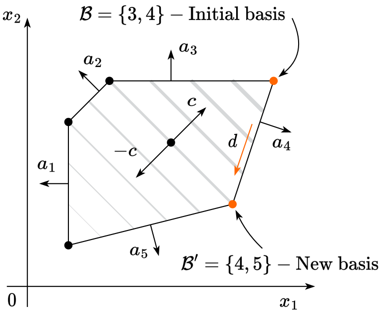


Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

$$\begin{cases} A_{\mathcal{B} \setminus \{k\}} d = 0 \\ a_k^T d = -1 \end{cases}$$

Changing basis

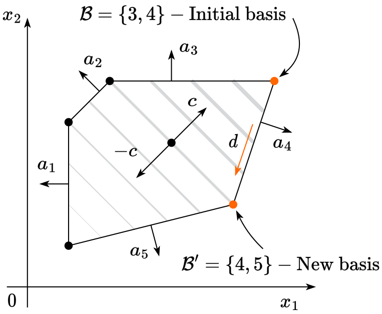


Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda^T_{\mathcal{B}} = c^T A^{-1}_{\mathcal{B}}$
- Let's assume, that $\lambda^k_{\mathcal{B}} > 0$. We'd like to drop k from the basis and form a new one:

$$
\begin{cases} A_{\mathcal{B} \setminus \{k\}} d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d
$$

Changing basis

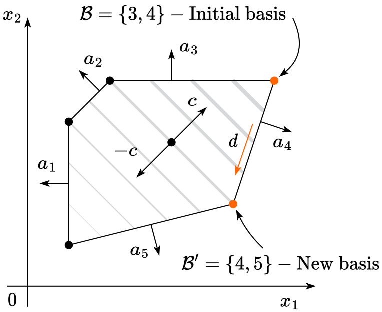


Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda^T_{\mathcal{B}} = c^T A^{-1}_{\mathcal{B}}$
- Let's assume, that $\lambda^k_{\mathcal{B}} > 0$. We'd like to drop k from the basis and form a new one:

$$
\begin{cases} A_{\mathcal{B} \setminus \{k\}} d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda^T_{\mathcal{B}} A_{\mathcal{B}} d
$$

Changing basis

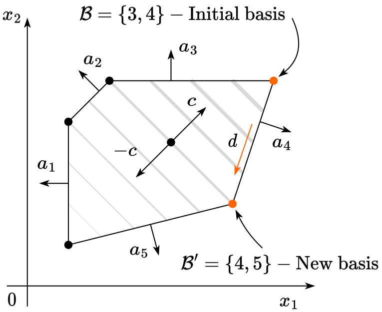


Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda^T_{\mathcal{B}} = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda^k_{\mathcal{B}} > 0$. We'd like to drop k from the basis and form a new one:

$$\begin{cases} A_{\mathcal{B} \setminus \{k\}} d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda^T_{\mathcal{B}} A_{\mathcal{B}} d = \sum_{i=1}^n \lambda^i_{\mathcal{B}} (A_{\mathcal{B}} d)^i$$

Changing basis

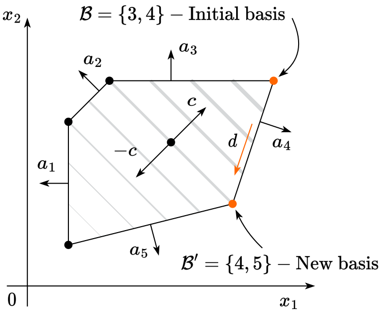


Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda^T_{\mathcal{B}} = c^T A^{-1}_{\mathcal{B}}$
- Let's assume, that $\lambda^k_{\mathcal{B}} > 0$. We'd like to drop k from the basis and form a new one:

$$\begin{cases} A_{\mathcal{B} \setminus \{k\}} d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda^T_{\mathcal{B}} A_{\mathcal{B}} d = \sum_{i=1}^n \lambda^i_{\mathcal{B}} (A_{\mathcal{B}} d)^i = -\lambda^k_{\mathcal{B}} < 0$$

Changing basis



Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

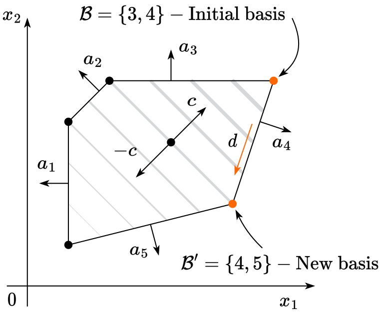
- Suppose, we have a basis $\mathcal{B}$: $\lambda^T_{\mathcal{B}} = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda^k_{\mathcal{B}} > 0$. We'd like to drop k from the basis and form a new one:

$$\begin{cases} A_{\mathcal{B}\setminus\{k\}} d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda^T_{\mathcal{B}} A_{\mathcal{B}} d = \sum_{i=1}^n \lambda^i_{\mathcal{B}} (A_{\mathcal{B}} d)^i = -\lambda^k_{\mathcal{B}} < 0$$

- For all $j \notin \mathcal{B}$ calculate the projection stepsize:

$$\mu_j = \frac{b_j - a_j^T x_{\mathcal{B}}}{a_j^T d}$$

Changing basis



Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

$$\begin{cases} A_{\mathcal{B}\setminus\{k\}} d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda_{\mathcal{B}}^T A_{\mathcal{B}} d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i (A_{\mathcal{B}} d)^i = -\lambda_{\mathcal{B}}^k < 0$$

- For all $j \notin \mathcal{B}$ calculate the projection stepsize:

$$\mu_j = \frac{b_j - a_j^T x_{\mathcal{B}}}{a_j^T d}$$

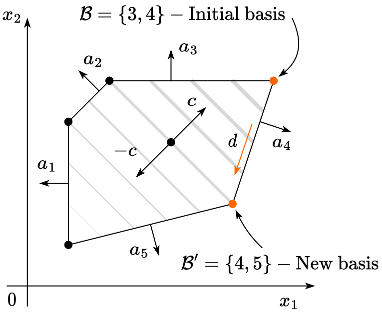
- Define the new vertex, that you will add to the new basis:

$$t = \arg\min_j \{\mu_j \mid \mu_j > 0\}$$

$$\mathcal{B}' = \mathcal{B}\setminus\{k\} \cup \{t\}$$

$$x_{\mathcal{B}'} = x_{\mathcal{B}} + \mu_t d = A_{\mathcal{B}'}^{-1} b_{\mathcal{B}'}$$

Changing basis



Suppose, some of the coefficients of $\lambda_{\mathcal{B}}$ are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis $\mathcal{B}$: $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

$$\begin{cases} A_{\mathcal{B}\setminus\{k\}} d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda_{\mathcal{B}}^T A_{\mathcal{B}} d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i (A_{\mathcal{B}} d)^i = -\lambda_{\mathcal{B}}^k < 0$$

- For all $j \notin \mathcal{B}$ calculate the projection stepsize:

$$\mu_j = \frac{b_j - a_j^T x_{\mathcal{B}}}{a_j^T d}$$

- Define the new vertex, that you will add to the new basis:

$$t = \text{arg}\min_j \{\mu_j \mid \mu_j > 0\}$$
$$\mathcal{B}' = \mathcal{B}\setminus\{k\} \cup \{t\}$$
$$x_{\mathcal{B}'} = x_{\mathcal{B}} + \mu_t d = A_{\mathcal{B}'}^{-1} b_{\mathcal{B}'}$$

- Note, that changing basis implies objective function decreasing

$$c^T x_{\mathcal{B}'} = c^T (x_{\mathcal{B}} + \mu_t d) = c^T x_{\mathcal{B}} + \mu_t c^T d$$

Finding an initial basic feasible solution

We aim to solve the following problem:

$$
\begin{split}
& \min_{x \in \mathbb{R}^n} c^{\top}x \\
\text{s.t. } & Ax \leq b
\end{split}
\tag{1}
$$

The proposed algorithm requires an initial basic feasible solution and corresponding basis.

Finding an initial basic feasible solution

We aim to solve the following problem:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } & Ax \leq b
\end{aligned}
\tag{1}
$$

The proposed algorithm requires an initial basic feasible solution and corresponding basis.

We start by reformulating the problem:

$$
\begin{aligned}
&\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\
\text{s.t. } & Ay - Az \leq b \\
& y \geq 0, z \geq 0
\end{aligned}
\tag{2}
$$

Finding an initial basic feasible solution

We aim to solve the following problem:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \; & c^\top x \\
\text{s.t. } & Ax \leq b
\end{aligned}
\tag{1}
$$

The proposed algorithm requires an initial basic feasible solution and corresponding basis.

We start by reformulating the problem:

$$
\begin{aligned}
\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} \; & c^\top (y - z) \\
\text{s.t. } & Ay - Az \leq b \\
& y \geq 0, z \geq 0
\end{aligned}
\tag{2}
$$

Given the solution of Problem 2 the solution of Problem 1 can be recovered and vice versa

$$
x = y - z \qquad \Leftrightarrow \qquad y_i = \max(x_i, 0), \quad z_i = \max(-x_i, 0)
$$

Now we will try to formulate new LP problem, which solution will be basic feasible point for Problem 2. Which means, that we firstly run Simplex method for Phase-1 problem and run Phase-2 problem with known starting point. Note, that basic feasible solution for Phase-1 should be somehow easily established.

Finding an initial basic feasible solution

$$
\begin{aligned}
\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} \; & c^\top (y - z) \\
\text{s.t. } & Ay - Az \leq b \qquad\qquad \text{(Phase-2 (Main LP))} \\
& y \geq 0, z \geq 0
\end{aligned}
$$

Finding an initial basic feasible solution

$$
\begin{aligned}
&\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\
\text{s.t. } & Ay - Az \le b \\
& y \ge 0, z \ge 0
\end{aligned}
\tag{Phase-2 (Main LP)}
$$

$$
\begin{aligned}
&\min_{\xi \in \mathbb{R}^m, y \in \mathbb{R}^n, z \in \mathbb{R}^n} \sum_{i=1}^m \xi_i \\
\text{s.t. } & Ay - Az \le b + \xi \\
& y \ge 0, z \ge 0, \xi \ge 0
\end{aligned}
\tag{Phase-1}
$$

Finding an initial basic feasible solution

$$
\begin{aligned}
&\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\
\text{s.t. } & Ay - Az \leq b \\
& y \geq 0, z \geq 0
\end{aligned}
\tag{Phase-2 (Main LP)}
$$

$$
\begin{aligned}
&\min_{\xi \in \mathbb{R}^m, y \in \mathbb{R}^n, z \in \mathbb{R}^n} \sum_{i=1}^m \xi_i \\
\text{s.t. } & Ay - Az \leq b + \xi \\
& y \geq 0, z \geq 0, \xi \geq 0
\end{aligned}
\tag{Phase-1}
$$

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
  Proof: trivial check.

Finding an initial basic feasible solution

$$
\begin{aligned}
&\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\
\text{s.t. } & Ay - Az \leq b \\
& y \geq 0, z \geq 0
\end{aligned}
\tag{Phase-2 (Main LP)}
$$

$$
\begin{aligned}
&\min_{\xi \in \mathbb{R}^m, y \in \mathbb{R}^n, z \in \mathbb{R}^n} \sum_{i=1}^m \xi_i \\
\text{s.t. } & Ay - Az \leq b + \xi \\
& y \geq 0, z \geq 0, \xi \geq 0
\end{aligned}
\tag{Phase-1}
$$

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
  Proof: trivial check.
- If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2.
  Proof: trivial check.

Finding an initial basic feasible solution

$$
\begin{aligned}
\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} \; & c^\top (y - z) \\
\text{s.t. } \; & Ay - Az \leq b \\
& y \geq 0, z \geq 0
\end{aligned}
\tag{Phase-2 (Main LP)}
$$

$$
\begin{aligned}
\min_{\xi \in \mathbb{R}^m, y \in \mathbb{R}^n, z \in \mathbb{R}^n} \; & \sum_{i=1}^m \xi_i \\
\text{s.t. } \; & Ay - Az \leq b + \xi \\
& y \geq 0, z \geq 0, \xi \geq 0
\end{aligned}
\tag{Phase-1}
$$

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
  Proof: trivial check.
- If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2.
  Proof: trivial check.

Finding an initial basic feasible solution

$$
\begin{aligned}
\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} \; & c^\top (y - z) \\
\text{s.t. } & Ay - Az \leq b \\
& y \geq 0, z \geq 0
\end{aligned}
\tag{Phase-2 (Main LP)}
$$

$$
\begin{aligned}
\min_{\xi \in \mathbb{R}^m, y \in \mathbb{R}^n, z \in \mathbb{R}^n} \; & \sum_{i=1}^{m} \xi_i \\
\text{s.t. } & Ay - Az \leq b + \xi \\
& y \geq 0, z \geq 0, \xi \geq 0
\end{aligned}
\tag{Phase-1}
$$

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
  Proof: trivial check.
- If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2.
  Proof: trivial check.

- Now we know, that if we can solve a Phase-1 problem then we will either find a starting point for the simplex method in the original method (if slacks are zero) or verify that the original problem was infeasible (if slacks are non-zero).

Finding an initial basic feasible solution

$$
\begin{aligned}
&\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z) \\
\text{s.t. } & Ay - Az \leq b \\
& y \geq 0, z \geq 0
\end{aligned}
\tag{Phase-2 (Main LP)}
$$

$$
\begin{aligned}
&\min_{\xi \in \mathbb{R}^m, y \in \mathbb{R}^n, z \in \mathbb{R}^n} \sum_{i=1}^m \xi_i \\
\text{s.t. } & Ay - Az \leq b + \xi \\
& y \geq 0, z \geq 0, \xi \geq 0
\end{aligned}
\tag{Phase-1}
$$

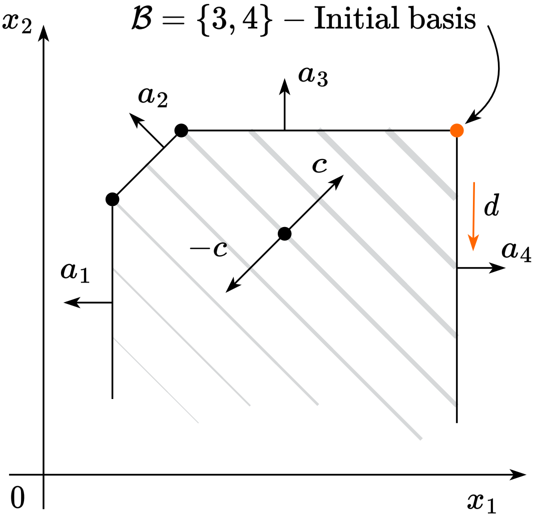
- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
  Proof: trivial check.
- If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2.
  Proof: trivial check.

- Now we know, that if we can solve a Phase-1 problem then we will either find a starting point for the simplex method in the original method (if slacks are zero) or verify that the original problem was infeasible (if slacks are non-zero).
- But how to solve Phase-1? It has basic feasible solution (the problem has $2n + m$ variables and the point below ensures $2n + m$ inequalities are satisfied as equalities (active).)

$$
z = 0 \quad y = 0 \quad \xi_i = \max(0, -b_i)
$$

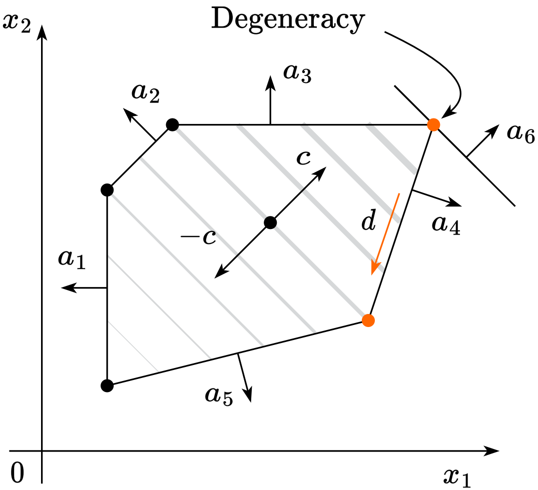
Convergence of the Simplex method

Unbounded budget set



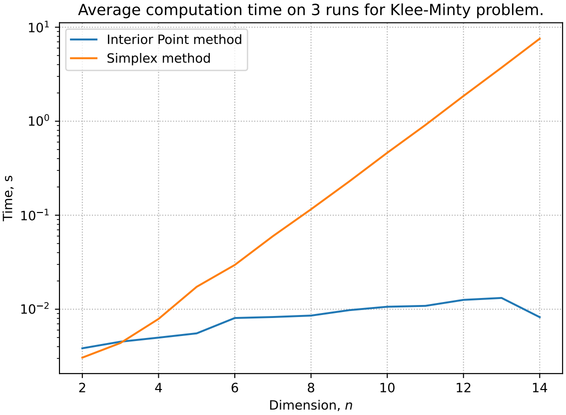
In this case, all μ_j will be negative.

Degeneracy



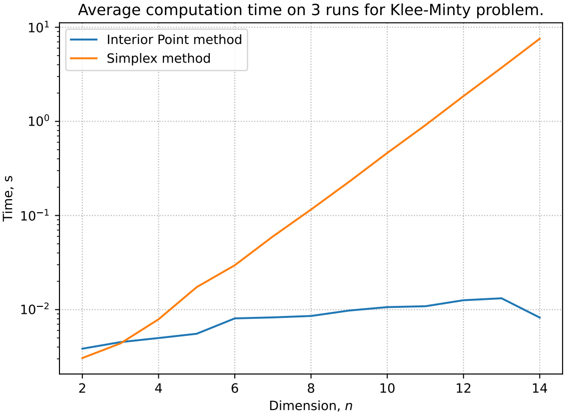
One needs to handle degenerate corners carefully. If no degeneracy exists, one can guarantee a monotonic decrease of the objective function on each iteration.

Exponential convergence



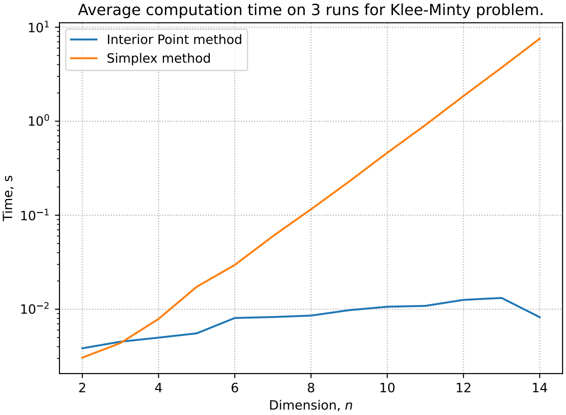
- A wide variety of applications could be formulated as linear programming.

Exponential convergence



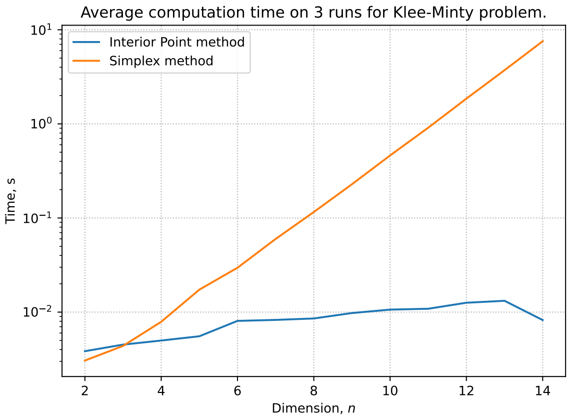
- A wide variety of applications could be formulated as linear programming.
- Simplex method is simple but could work exponentially long.

Exponential convergence



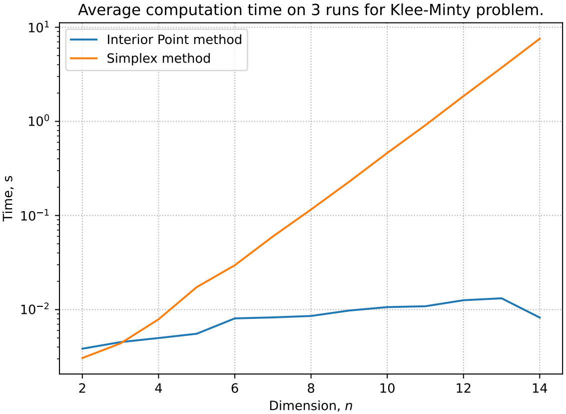
- A wide variety of applications could be formulated as linear programming.
- Simplex method is simple but could work exponentially long.
- Khachiyan's ellipsoid method (1979) is the first to be proven to run at polynomial complexity for LPs. However, it is usually slower than simplex in real problems.

Exponential convergence



- A wide variety of applications could be formulated as linear programming.
- Simplex method is simple but could work exponentially long.
- Khachiyan's ellipsoid method (1979) is the first to be proven to run at polynomial complexity for LPs. However, it is usually slower than simplex in real problems.
- Major breakthrough - Narendra Karmarkar's method for solving LP (1984) using interior point method.

Exponential convergence



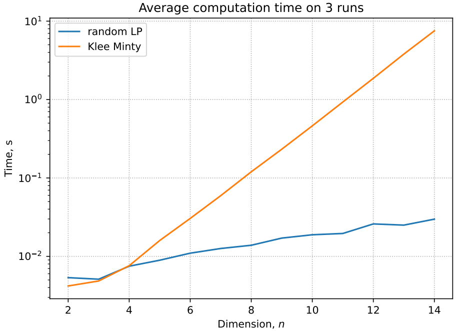
- A wide variety of applications could be formulated as linear programming.
- Simplex method is simple but could work exponentially long.
- Khachiyan's ellipsoid method (1979) is the first to be proven to run at polynomial complexity for LPs. However, it is usually slower than simplex in real problems.
- Major breakthrough - Narendra Karmarkar's method for solving LP (1984) using interior point method.
- Interior point methods are the last word in this area. However, good implementations of simplex-based methods and interior point methods are similar for routine applications of linear programming.

Klee Minty example

Since the number of edge points is finite, the algorithm should converge (except for some degenerate cases, which are not covered here). However, the convergence could be exponentially slow, due to the high number of edges. There is the following iconic example when the simplex method should perform exactly all vertexes.

In the following problem, the simplex method needs to check $2^n - 1$ vertexes with $x_0 = 0$.

$$
\begin{aligned}
& \max_{x \in \mathbb{R}^n} 2^{n-1}x_1 + 2^{n-2}x_2 + \dots + 2x_{n-1} + x_n \\
\text{s.t. } & x_1 \leq 5 \\
& 4x_1 + x_2 \leq 25 \\
& 8x_1 + 4x_2 + x_3 \leq 125 \\
& \ldots \\
& 2^n x_1 + 2^{n-1}x_2 + 2^{n-2}x_3 + \ldots + x_n \leq 5^n \\
& x \geq 0
\end{aligned}
$$



Duality in Linear Programming

Duality

Primal problem:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } & Ax = b \\
& x_i \geq 0, \; i = 1, \dots, n
\end{aligned}
\tag{3}
$$

Duality

Primal problem:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } & Ax = b \\
& x_i \geq 0, \; i = 1, \dots, n
\end{aligned}
\tag{3}
$$

KKT for optimal x^*, ν^*, λ^*:

$$
\begin{aligned}
&L(x, \nu, \lambda) = c^T x + \nu^T (Ax - b) - \lambda^T x \\
&-A^T \nu^* + \lambda^* = c \\
&Ax^* = b \\
&x^* \succeq 0 \\
&\lambda^* \succeq 0 \\
&\lambda_i^* x_i^* = 0
\end{aligned}
$$

Duality

Primal problem:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \; & c^\top x \\
\text{s.t. } \; & Ax = b \\
& x_i \geq 0, \; i = 1, \dots, n
\end{aligned}
\tag{3}
$$

KKT for optimal x^*, ν^*, λ^*:

$$
\begin{aligned}
& L(x, \nu, \lambda) = c^T x + \nu^T (Ax - b) - \lambda^T x \\
& -A^T \nu^* + \lambda^* = c \\
& Ax^* = b \\
& x^* \succeq 0 \\
& \lambda^* \succeq 0 \\
& \lambda_i^* x_i^* = 0
\end{aligned}
$$

Has the following dual:

$$
\begin{aligned}
\max_{\nu \in \mathbb{R}^m} \; & -b^\top \nu \\
\text{s.t. } \; & -A^T \nu \preceq c
\end{aligned}
\tag{4}
$$

Find the dual problem to the problem above (it should be the original LP). Also, write down KKT for the dual problem, to ensure, they are identical to the primal KKT.

Strong duality in linear programming

(i) If either problem Equation 3 or Equation 4 has a (finite) solution, then so does the other, and the objective values are equal.

Strong duality in linear programming

(i) If either problem Equation 3 or Equation 4 has a (finite) solution, then so does the other, and the objective values are equal.

(ii) If either problem Equation 3 or Equation 4 is unbounded, then the other problem is infeasible.

Strong duality in linear programming

(i) If either problem Equation 3 or Equation 4 has a (finite) solution, then so does the other, and the objective values are equal.

(ii) If either problem Equation 3 or Equation 4 is unbounded, then the other problem is infeasible.

Strong duality in linear programming

(i) If either problem Equation 3 or Equation 4 has a (finite) solution, then so does the other, and the objective values are equal.

(ii) If either problem Equation 3 or Equation 4 is unbounded, then the other problem is infeasible.

PROOF. For (i), suppose that Equation 3 has a finite optimal solution x^*. It follows from KKT that there are optimal vectors λ^* and ν^* such that (x^*, ν^*, λ^*) satisfies KKT. We noted above that KKT for Equation 3 and Equation 4 are equivalent. Moreover, $c^T x^* = (-A^T \nu^* + \lambda^*)^T x^* = -(\nu^*)^T A x^* = -b^T \nu^*$, as claimed.

A symmetric argument holds if we start by assuming that the dual problem Equation 4 has a solution.

Strong duality in linear programming

(i) If either problem Equation 3 or Equation 4 has a (finite) solution, then so does the other, and the objective values are equal.

(ii) If either problem Equation 3 or Equation 4 is unbounded, then the other problem is infeasible.

PROOF. For (i), suppose that Equation 3 has a finite optimal solution x^*. It follows from KKT that there are optimal vectors λ^* and ν^* such that (x^*, ν^*, λ^*) satisfies KKT. We noted above that KKT for Equation 3 and Equation 4 are equivalent. Moreover, $c^T x^* = (-A^T \nu^* + \lambda^*)^T x^* = -(\nu^*)^T A x^* = -b^T \nu^*$, as claimed.

A symmetric argument holds if we start by assuming that the dual problem Equation 4 has a solution.

To prove (ii), suppose that the primal is unbounded, that is, there is a sequence of points x_k, $k = 1, 2, 3, \dots$ such that

$$c^T x_k \downarrow -\infty, \quad A x_k = b, \quad x_k \geq 0.$$

Strong duality in linear programming

(i) If either problem Equation 3 or Equation 4 has a (finite) solution, then so does the other, and the objective values are equal.

(ii) If either problem Equation 3 or Equation 4 is unbounded, then the other problem is infeasible.

PROOF. For (i), suppose that Equation 3 has a finite optimal solution x^*. It follows from KKT that there are optimal vectors λ^* and ν^* such that (x^*, ν^*, λ^*) satisfies KKT. We noted above that KKT for Equation 3 and Equation 4 are equivalent. Moreover, $c^T x^* = (-A^T \nu^* + \lambda^*)^T x^* = -(\nu^*)^T A x^* = -b^T \nu^*$, as claimed.

A symmetric argument holds if we start by assuming that the dual problem Equation 4 has a solution.

To prove (ii), suppose that the primal is unbounded, that is, there is a sequence of points x_k, $k = 1, 2, 3, \ldots$ such that

$$c^T x_k \downarrow -\infty, \quad Ax_k = b, \quad x_k \geq 0.$$

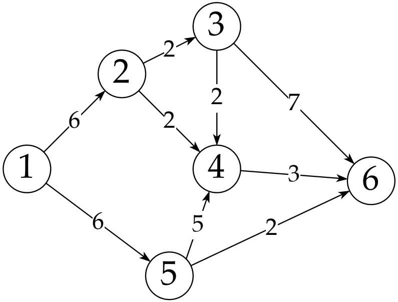
Suppose too that the dual Equation 4 is feasible, that is, there exists a vector $\bar{\nu}$ such that $-A^T \bar{\nu} \leq c$. From the latter inequality together with $x_k \geq 0$, we have that $-\bar{\nu}^T A x_k \leq c^T x_k$, and therefore

$$-\bar{\nu}^T b = -\bar{\nu}^T A x_k \leq c^T x_k \downarrow -\infty,$$

yielding a contradiction. Hence, the dual must be infeasible. A similar argument can be used to show that the unboundedness of the dual implies the infeasibility of the primal.

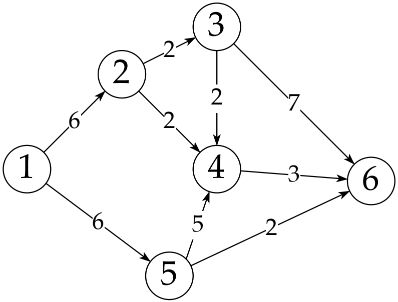
Max-flow min-cut

Max-flow problem example



The nodes are routers, the edges are communications links; associated with each node is a capacity — node 1 can communicate to node 2 at as much as 6 Mbps, node 2 can communicate to node 4 at upto 2 Mbps, etc.

Max-flow problem example

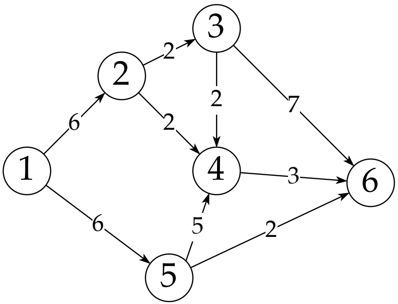


The nodes are routers, the edges are communications links; associated with each node is a capacity — node 1 can communicate to node 2 at as much as 6 Mbps, node 2 can communicate to node 4 at upto 2 Mbps, etc.

Question:

- A network of nodes and edges represents communication links, each with a specified capacity.

Max-flow problem example

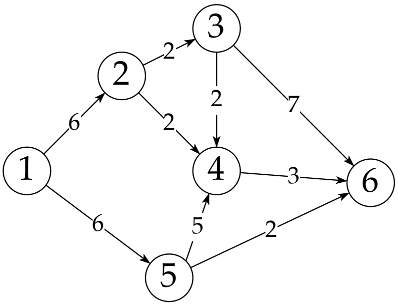


The nodes are routers, the edges are communications links; associated with each node is a capacity — node 1 can communicate to node 2 at as much as 6 Mbps, node 2 can communicate to node 4 at upto 2 Mbps, etc.

Question:

- A network of nodes and edges represents communication links, each with a specified capacity.
- Example: Can node 1 (source) communicate with node 6 (sink) at 6 Mbps? 12 Mbps? What is the maximum rate?

Max-flow problem example

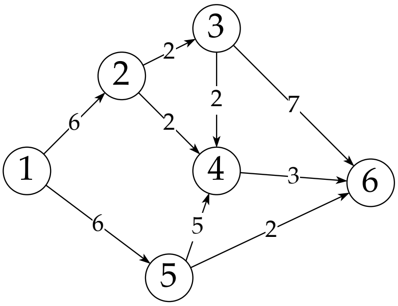


The nodes are routers, the edges are communications links; associated with each node is a capacity — node 1 can communicate to node 2 at as much as 6 Mbps, node 2 can communicate to node 4 at upto 2 Mbps, etc.

Question:

- A network of nodes and edges represents communication links, each with a specified capacity.
- Example: Can node 1 (source) communicate with node 6 (sink) at 6 Mbps? 12 Mbps? What is the maximum rate?

Max-flow problem example



The nodes are routers, the edges are communications links; associated with each node is a capacity — node 1 can communicate to node 2 at as much as 6 Mbps, node 2 can communicate to node 4 at upto 2 Mbps, etc.

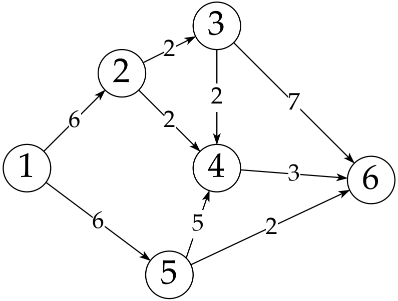
Question:

- A network of nodes and edges represents communication links, each with a specified capacity.
- Example: Can node 1 (source) communicate with node 6 (sink) at 6 Mbps? 12 Mbps? What is the maximum rate?

Capacity Matrix:

$$
C = \begin{bmatrix} 0 & 6 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

Max-flow problem example



The nodes are routers, the edges are communications links; associated with each node is a capacity — node 1 can communicate to node 2 at as much as 6 Mbps, node 2 can communicate to node 4 at upto 2 Mbps, etc.

Question:

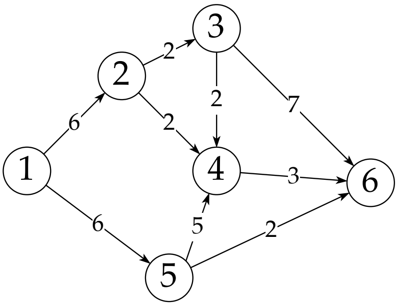
- A network of nodes and edges represents communication links, each with a specified capacity.
- Example: Can node 1 (source) communicate with node 6 (sink) at 6 Mbps? 12 Mbps? What is the maximum rate?

Capacity Matrix:

$$
C = \begin{bmatrix} 0 & 6 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

Flow Matrix: $X[i, j]$ represents flow from node i to node j.

Max-flow problem example



The nodes are routers, the edges are communications links; associated with each node is a capacity — node 1 can communicate to node 2 at as much as 6 Mbps, node 2 can communicate to node 4 at upto 2 Mbps, etc.

Question:

- A network of nodes and edges represents communication links, each with a specified capacity.
- Example: Can node 1 (source) communicate with node 6 (sink) at 6 Mbps? 12 Mbps? What is the maximum rate?

Capacity Matrix:

$$C = \begin{bmatrix} 0 & 6 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

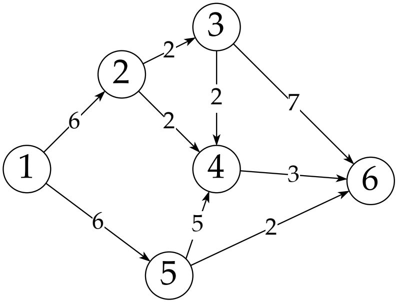
Flow Matrix: $X[i, j]$ represents flow from node i to node j.

Constraints:

$$0 \preceq X \qquad X \preceq C$$

Flow Conservation: $$\sum_{j=2}^{N} X(i,j) = \sum_{k=1}^{N-1} X(k,i), \; i = 2, \ldots, N-1$$

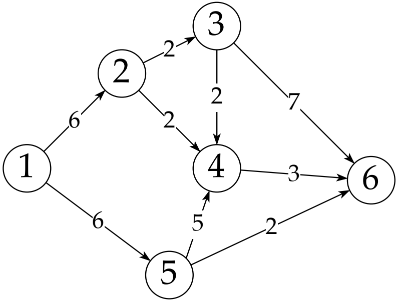
Max-flow problem example



Given the setup, when everything, that is produced by source will go to the sink. the flow of the network, is simply the sum of everything coming out of the source:

$$\sum_{i=2}^{N} X(1, i) \tag{Flow}$$

Max-flow problem example



$$
\begin{aligned}
\text{maximize } & \langle X, S \rangle \\
\text{s.t. } & -X \preceq 0 \\
& X \preceq C \\
& \langle X, L_n \rangle = 0, \; n = 2, \ldots, N-1,
\end{aligned}
\qquad \text{(Max-Flow Problem)}
$$

L_n consists of a single column (n) of ones (except for the last row) minus a single row (also n) of ones (except for the first column).

$$
S = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & -1 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.
$$

Given the setup, when everything, that is produced by source will go to the sink. the flow of the network, is simply the sum of everything coming out of the source:

$$
\sum_{i=2}^{N} X(1, i) \qquad \text{(Flow)}
$$

Deriving dual to the Max-flow

Deriving dual to the Max-flow

$$
\begin{aligned}
&\text{minimize } \langle \Lambda, C \rangle \\
&\qquad \Lambda, \nu \\
&\text{s.t. } \Lambda + Q \succeq S \\
&\qquad \Lambda \succeq 0
\end{aligned}
\tag{Max-Flow Dual Problem}
$$

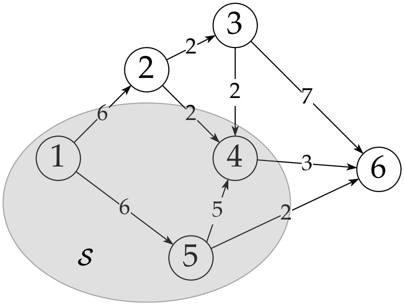
where

$$
Q = \begin{bmatrix}
0 & \nu_2 & \nu_3 & \cdots & \nu_{N-1} & 0 \\
0 & 0 & \nu_3 - \nu_2 & \cdots & \nu_{N-1} - \nu_2 & -\nu_2 \\
0 & \nu_2 - \nu_3 & 0 & \cdots & \nu_{N-1} - \nu_3 & -\nu_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \nu_2 - \nu_{N-1} & \nu_3 - \nu_{N-1} & \cdots & 0 & -\nu_{N-1} \\
0 & 0 & 0 & \cdots & 0 & 0
\end{bmatrix}.
$$

Min-cut problem example

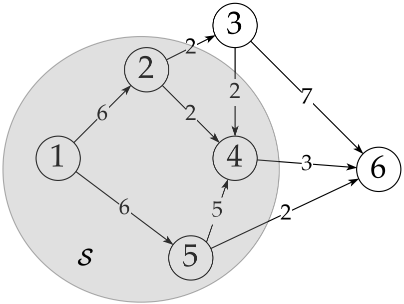
A cut of the network separates the vertices into two sets: one containing the source (we call this set $\mathcal{S}$, and one containing the sink. The capacity of the cut is the total value of the edges coming out of $\mathcal{S}$ — we are separating the sets by "cutting off the flow" along these edges.

$\mathcal{S} = \{1, 4, 5\}$



The edges in the cut are $1 \to 2, 4 \to 6$, and $5 \to 6$ the capacity of this cut is $6 + 3 + 2 = 11$.

$\mathcal{S} = \{1, 2, 4, 5\}$



The edges in the cut are $2 \to 3, 4 \to 6$, and $5 \to 6$ the capacity of this cut is $2 + 3 + 2 = 7$.

Min-cut is the dual to max-flow

What is the minimum value of the smallest cut? We will argue that it is same as the optimal value of the solution d^* of the dual program (Max-Flow Dual Problem).

Min-cut is the dual to max-flow

What is the minimum value of the smallest cut? We will argue that it is same as the optimal value of the solution d^* of the dual program (Max-Flow Dual Problem).

First, suppose that $\mathcal{S}$ is a valid cut. From $\mathcal{S}$, we can easily find a dual feasible point that matches its capacity: for $n = 1, \ldots, N$, take

$$\nu_n = \begin{cases} 1, & n \in \mathcal{S}, \\ 0, & n \notin \mathcal{S}, \end{cases} \qquad \text{and} \qquad \lambda_{i,j} = \begin{cases} \max(\nu_i - \nu_j, 0), & i \neq 1,\ j \neq N, \\ 1 - \nu_j, & i = 1, \\ \nu_i, & j = N. \end{cases}$$

Min-cut is the dual to max-flow

What is the minimum value of the smallest cut? We will argue that it is same as the optimal value of the solution d^* of the dual program (Max-Flow Dual Problem).

First, suppose that $\mathcal{S}$ is a valid cut. From $\mathcal{S}$, we can easily find a dual feasible point that matches its capacity: for $n = 1, \ldots, N$, take

$$\nu_n = \begin{cases} 1, & n \in \mathcal{S}, \\ 0, & n \notin \mathcal{S}, \end{cases} \qquad \text{and} \qquad \lambda_{i,j} = \begin{cases} \max(\nu_i - \nu_j, 0), & i \neq 1,\ j \neq N, \\ 1 - \nu_j, & i = 1, \\ \nu_i, & j = N. \end{cases}$$

Notice that these choices obey the constraints in the dual, and that $\lambda_{i,j}$ will be 1 if $i \to j$ is cut, and 0 otherwise, so

$$\text{capacity}(S) = \sum_{i,j} \lambda_{i,j} C_{i,j}.$$

Min-cut is the dual to max-flow

What is the minimum value of the smallest cut? We will argue that it is same as the optimal value of the solution d^* of the dual program (Max-Flow Dual Problem).

First, suppose that $\mathcal{S}$ is a valid cut. From $\mathcal{S}$, we can easily find a dual feasible point that matches its capacity: for $n = 1, \ldots, N$, take

$$\nu_n = \begin{cases} 1, & n \in \mathcal{S}, \\ 0, & n \notin \mathcal{S}, \end{cases} \qquad \text{and} \qquad \lambda_{i,j} = \begin{cases} \max(\nu_i - \nu_j, 0), & i \neq 1,\ j \neq N, \\ 1 - \nu_j, & i = 1, \\ \nu_i, & j = N. \end{cases}$$

Notice that these choices obey the constraints in the dual, and that $\lambda_{i,j}$ will be 1 if $i \to j$ is cut, and 0 otherwise, so

$$\text{capacity}(S) = \sum_{i,j} \lambda_{i,j} C_{i,j}.$$

Every cut is feasible, so

$$d^\star \leq \text{MINCUT}.$$

Min-cut is the dual to max-flow

Now we show that for every solution ν^*, λ^* of the dual, there is a cut that has a capacity at most d^*. We generate a cut *at random*, and then show that the expected value of the capacity of the cut is less than d^* — this means there must be at least one with a capacity of d^* or less.

Min-cut is the dual to max-flow

Now we show that for every solution ν^*, λ^* of the dual, there is a cut that has a capacity at most d^*. We generate a cut *at random*, and then show that the expected value of the capacity of the cut is less than d^* — this means there must be at least one with a capacity of d^* or less.

Let Z be a uniform random variable on $[0,1]$. Along with $\lambda^*, \nu_2^*, \ldots, \nu_{N-1}^*$ generated by solving (Max-Flow Dual Problem), take $\nu_1 = 1$ and $\nu_N = 0$. Create a cut $\mathcal{S}$ with the rule:

$$\text{if } \nu_n^* > Z, \text{ then take } n \in \mathcal{S}.$$

. . . The probability that a particular edge $i \to j$ is in this cut is

$$\begin{aligned}
P(i \in \mathcal{S}, j \notin \mathcal{S}) &= P\left(\nu_j^\star \le Z \le \nu_i^\star\right) \\
&\le \begin{cases} \max(\nu_i^\star - \nu_j^\star, 0), & 2 \le i, j \le N-1, \\ 1 - \nu_j^\star, & i = 1;\ j = 2, \ldots, N-1, \\ \nu_i^\star, & i = 2, \ldots, N-1;\ j = N, \\ 1, & i = 1;\ j = N. \end{cases} \\
&\le \lambda_{i,j}^\star,
\end{aligned}$$

Min-cut is the dual to max-flow

The last inequality follows simply from the constraints in the dual program (Max-Flow Dual Problem). This cut is random, so its capacity is a random variable, and its expectation is

$$\begin{aligned}
\mathbb{E}[\text{capacity}(\mathcal{S})] &= \sum_{i,j} C_{i,j} P(i \in \mathcal{S}, j \notin \mathcal{S}) \\
&\leq \sum_{i,j} C_{i,j} \lambda^{\star}_{i,j} = d^{\star}.
\end{aligned}$$

Min-cut is the dual to max-flow

The last inequality follows simply from the constraints in the dual program (Max-Flow Dual Problem). This cut is random, so its capacity is a random variable, and its expectation is

$$
\begin{aligned}
\mathbb{E}[\text{capacity}(\mathcal{S})] &= \sum_{i,j} C_{i,j} P(i \in \mathcal{S}, j \notin \mathcal{S}) \\
&\leq \sum_{i,j} C_{i,j} \lambda^\star_{i,j} = d^\star.
\end{aligned}
$$

Thus there must be a cut whose capacity is at most $d^\star$. This establishes that

$$
\text{MINCUT} \leq d^\star.
$$

Min-cut is the dual to max-flow

The last inequality follows simply from the constraints in the dual program (Max-Flow Dual Problem). This cut is random, so its capacity is a random variable, and its expectation is

$$\mathbb{E}[\text{capacity}(\mathcal{S})] = \sum_{i,j} C_{i,j} P(i \in \mathcal{S}, j \notin \mathcal{S}) \leq \sum_{i,j} C_{i,j} \lambda^\star_{i,j} = d^\star.$$

Thus there must be a cut whose capacity is at most $d^\star$. This establishes that

$$\text{MINCUT} \leq d^\star.$$

Combining these two facts of course means that

$$d^\star = \text{MINCUT} = \text{MAXFLOW} = p^\star,$$

where $p^\star$ is the solution of the primal, and equality follows from strong duality for linear programming.

Min-cut is the dual to max-flow

The last inequality follows simply from the constraints in the dual program (Max-Flow Dual Problem). This cut is random, so its capacity is a random variable, and its expectation is

$$\begin{aligned}\mathbb{E}[\text{capacity}(\mathcal{S})] &= \sum_{i,j} C_{i,j} P(i \in \mathcal{S}, j \notin \mathcal{S}) \\ &\leq \sum_{i,j} C_{i,j} \lambda^\star_{i,j} = d^\star.\end{aligned}$$

Thus there must be a cut whose capacity is at most $d^\star$. This establishes that

$$\text{MINCUT} \leq d^\star.$$

Combining these two facts of course means that

$$d^\star = \text{MINCUT} = \text{MAXFLOW} = p^\star,$$

where $p^\star$ is the solution of the primal, and equality follows from strong duality for linear programming.

Max-flow min-cut theorem.

The maximum value of an s-t flow is equal to the minimum capacity over all s-t cuts.

Sensitivity analysis

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{split}
& f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } & f_i(x) \leq 0, \; i = 1, \ldots, m \\
& h_i(x) = 0, \; i = 1, \ldots, p
\end{split}
\tag{P}
$$

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{split}
& f_0(x) \to \min\limits_{x \in \mathbb{R}^n}\\
\text{s.t. } & f_i(x) \leq 0, \; i = 1,\ldots,m\\
& h_i(x) = 0, \; i = 1,\ldots,p
\end{split} \tag{P}
$$

To the perturbed version of it:

$$
\begin{split}
& f_0(x) \to \min\limits_{x \in \mathbb{R}^n}\\
\text{s.t. } & f_i(x) \leq u_i, \; i = 1,\ldots,m\\
& h_i(x) = v_i, \; i = 1,\ldots,p
\end{split} \tag{Per}
$$

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{aligned}
& f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } & f_i(x) \leq 0, \; i = 1, \ldots, m \\
& h_i(x) = 0, \; i = 1, \ldots, p
\end{aligned}
\tag{P}
$$

To the perturbed version of it:

$$
\begin{aligned}
& f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } & f_i(x) \leq u_i, \; i = 1, \ldots, m \\
& h_i(x) = v_i, \; i = 1, \ldots, p
\end{aligned}
\tag{Per}
$$

Note, that we still have the only variable $x \in \mathbb{R}^n$, while treating $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ as parameters. It is obvious, that Per$(u, v) \to$ P if $u = 0, v = 0$. We will denote the optimal value of Per as $p^*(u, v)$, while the optimal value of the original problem P is just p^*. One can immediately say, that $p^*(u, v) = p^*$.

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{aligned}
&f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } &f_i(x) \leq 0, \; i = 1, \ldots, m \\
&h_i(x) = 0, \; i = 1, \ldots, p
\end{aligned}
\tag{P}
$$

To the perturbed version of it:

$$
\begin{aligned}
&f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } &f_i(x) \leq u_i, \; i = 1, \ldots, m \\
&h_i(x) = v_i, \; i = 1, \ldots, p
\end{aligned}
\tag{Per}
$$

Note, that we still have the only variable $x \in \mathbb{R}^n$, while treating $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ as parameters. It is obvious, that Per$(u, v) \to$ P if $u = 0, v = 0$. We will denote the optimal value of Per as $p^*(u, v)$, while the optimal value of the original problem P is just p^*. One can immediately say, that $p^*(u, v) = p^*$.

Speaking of the value of some i-th constraint we can say, that

- $u_i = 0$ leaves the original problem

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{aligned}
& f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } & f_i(x) \leq 0, \ i = 1, \dots, m \\
& h_i(x) = 0, \ i = 1, \dots, p
\end{aligned}
\tag{P}
$$

To the perturbed version of it:

$$
\begin{aligned}
& f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } & f_i(x) \leq u_i, \ i = 1, \dots, m \\
& h_i(x) = v_i, \ i = 1, \dots, p
\end{aligned}
\tag{Per}
$$

Note, that we still have the only variable $x \in \mathbb{R}^n$, while treating $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ as parameters. It is obvious, that Per$(u, v) \to$ P if $u = 0, v = 0$. We will denote the optimal value of Per as $p^*(u, v)$, while the optimal value of the original problem P is just p^*. One can immediately say, that $p^*(u, v) = p^*$.

Speaking of the value of some i-th constraint we can say, that

- $u_i = 0$ leaves the original problem
- $u_i > 0$ means that we have relaxed the inequality

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{aligned}
& f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } & f_i(x) \leq 0, \; i = 1, \ldots, m \\
& h_i(x) = 0, \; i = 1, \ldots, p
\end{aligned}
\tag{P}
$$

To the perturbed version of it:

$$
\begin{aligned}
& f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } & f_i(x) \leq u_i, \; i = 1, \ldots, m \\
& h_i(x) = v_i, \; i = 1, \ldots, p
\end{aligned}
\tag{Per}
$$

Note, that we still have the only variable $x \in \mathbb{R}^n$, while treating $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ as parameters. It is obvious, that Per$(u, v) \to$ P if $u = 0, v = 0$. We will denote the optimal value of Per as $p^*(u, v)$, while the optimal value of the original problem P is just p^*. One can immediately say, that $p^*(u, v) = p^*$.

Speaking of the value of some i-th constraint we can say, that

- $u_i = 0$ leaves the original problem
- $u_i > 0$ means that we have relaxed the inequality
- $u_i < 0$ means that we have tightened the constraint

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{aligned}
&f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } &f_i(x) \leq 0, \; i = 1, \ldots, m \\
&h_i(x) = 0, \; i = 1, \ldots, p
\end{aligned}
\tag{P}
$$

To the perturbed version of it:

$$
\begin{aligned}
&f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } &f_i(x) \leq u_i, \; i = 1, \ldots, m \\
&h_i(x) = v_i, \; i = 1, \ldots, p
\end{aligned}
\tag{Per}
$$

Note, that we still have the only variable $x \in \mathbb{R}^n$, while treating $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ as parameters. It is obvious, that Per$(u, v) \to$ P if $u = 0, v = 0$. We will denote the optimal value of Per as $p^*(u, v)$, while the optimal value of the original problem P is just p^*. One can immediately say, that $p^*(u, v) = p^*$.

Speaking of the value of some i-th constraint we can say, that

- $u_i = 0$ leaves the original problem
- $u_i > 0$ means that we have relaxed the inequality
- $u_i < 0$ means that we have tightened the constraint

Sensitivity analysis

Let us switch from the original optimization problem

$$
\begin{aligned}
&f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } &f_i(x) \leq 0, \; i = 1, \ldots, m \\
&h_i(x) = 0, \; i = 1, \ldots, p
\end{aligned}
\tag{P}
$$

To the perturbed version of it:

$$
\begin{aligned}
&f_0(x) \to \min_{x \in \mathbb{R}^n} \\
\text{s.t. } &f_i(x) \leq u_i, \; i = 1, \ldots, m \\
&h_i(x) = v_i, \; i = 1, \ldots, p
\end{aligned}
\tag{Per}
$$

Note, that we still have the only variable $x \in \mathbb{R}^n$, while treating $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ as parameters. It is obvious, that Per$(u, v) \to$ P if $u = 0, v = 0$. We will denote the optimal value of Per as $p^*(u, v)$, while the optimal value of the original problem P is just p^*. One can immediately say, that $p^*(u, v) = p^*$.

Speaking of the value of some i-th constraint we can say, that

- $u_i = 0$ leaves the original problem
- $u_i > 0$ means that we have relaxed the inequality
- $u_i < 0$ means that we have tightened the constraint

One can even show, that when P is convex optimization problem, $p^*(u, v)$ is a convex function.

Sensitivity analysis

Suppose, that strong duality holds for the orriginal problem and suppose, that x is any feasible point for the perturbed problem:

$$p^*(0,0) = p^* = d^* = g(\lambda^*, \nu^*) \leq$$

Sensitivity analysis

Suppose, that strong duality holds for the orriginal problem and suppose, that x is any feasible point for the perturbed problem:

$$p^*(0,0) = p^* = d^* = g(\lambda^*, \nu^*) \leq$$
$$\leq L(x, \lambda^*, \nu^*) =$$

Sensitivity analysis

Suppose, that strong duality holds for the orriginal problem and suppose, that x is any feasible point for the perturbed problem:

$$
\begin{split}
p^*(0,0) = p^* = d^* = g(\lambda^*, \nu^*) \leq \\
\leq L(x, \lambda^*, \nu^*) = \\
= f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \leq
\end{split}
$$

Sensitivity analysis

Suppose, that strong duality holds for the orriginal problem and suppose, that x is any feasible point for the perturbed problem:

$$
\begin{aligned}
p^*(0,0) &= p^* = d^* = g(\lambda^*, \nu^*) \leq \\
&\leq L(x, \lambda^*, \nu^*) = \\
&= f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \leq \\
&\leq f_0(x) + \sum_{i=1}^m \lambda_i^* u_i + \sum_{i=1}^p \nu_i^* v_i
\end{aligned}
$$

Sensitivity analysis

Suppose, that strong duality holds for the orriginal problem and suppose, that x is any feasible point for the perturbed problem:

$$
\begin{aligned}
p^*(0,0) = p^* = d^* = g(\lambda^*, \nu^*) \leq \\
&\leq L(x, \lambda^*, \nu^*) = \\
&= f_0(x) + \sum_{i=1}^{m} \lambda_i^* f_i(x) + \sum_{i=1}^{p} \nu_i^* h_i(x) \leq \\
&\leq f_0(x) + \sum_{i=1}^{m} \lambda_i^* u_i + \sum_{i=1}^{p} \nu_i^* v_i
\end{aligned}
$$

Sensitivity analysis

Suppose, that strong duality holds for the orriginal problem and suppose, that x is any feasible point for the perturbed problem:

$$
\begin{split}
p^*(0,0) = p^* = d^* = g(\lambda^*, \nu^*) \leq \\
\leq L(x, \lambda^*, \nu^*) = \\
= f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \leq \\
\leq f_0(x) + \sum_{i=1}^m \lambda_i^* u_i + \sum_{i=1}^p \nu_i^* v_i
\end{split}
$$

Which means

$$
f_0(x) \geq p^*(0,0) - {\lambda^*}^T u - {\nu^*}^T v
$$

Sensitivity analysis

Suppose, that strong duality holds for the orriginal problem and suppose, that x is any feasible point for the perturbed problem:

$$
\begin{aligned}
p^*(0,0) = p^* = d^* = g(\lambda^*, \nu^*) \leq \\
\leq L(x, \lambda^*, \nu^*) = \\
= f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \leq \\
\leq f_0(x) + \sum_{i=1}^m \lambda_i^* u_i + \sum_{i=1}^p \nu_i^* v_i
\end{aligned}
$$

Which means

$$
f_0(x) \geq p^*(0,0) - {\lambda^*}^T u - {\nu^*}^T v
$$

And taking the optimal x for the perturbed problem, we have:

$$
p^*(u,v) \geq p^*(0,0) - {\lambda^*}^T u - {\nu^*}^T v \tag{5}
$$

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or
  - If $\nu_i^\star$ is large and negative and $v_i > 0$ is selected,

  then in either scenario, the optimal value $p^\star(u, v)$ is expected to increase greatly.

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or
  - If $\nu_i^\star$ is large and negative and $v_i > 0$ is selected,

  then in either scenario, the optimal value $p^\star(u, v)$ is expected to increase greatly.
- **Consequences of Loosening a Constraint (Small $\lambda_i^\star$)**:
  If the Lagrange multiplier $\lambda_i^\star$ for the ith constraint is relatively small, and the constraint is loosened (choosing $u_i > 0$), it is anticipated that the optimal value $p^\star(u, v)$ will not significantly decrease.

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or
  - If $\nu_i^\star$ is large and negative and $v_i > 0$ is selected,

  then in either scenario, the optimal value $p^\star(u, v)$ is expected to increase greatly.
- **Consequences of Loosening a Constraint (Small $\lambda_i^\star$)**:
  If the Lagrange multiplier $\lambda_i^\star$ for the ith constraint is relatively small, and the constraint is loosened (choosing $u_i > 0$), it is anticipated that the optimal value $p^\star(u, v)$ will not significantly decrease.
- **Outcomes of Tiny Adjustments in Constraints with Small $\nu_i^\star$**:

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or
  - If $\nu_i^\star$ is large and negative and $v_i > 0$ is selected,

  then in either scenario, the optimal value $p^\star(u, v)$ is expected to increase greatly.
- **Consequences of Loosening a Constraint (Small $\lambda_i^\star$)**:
  If the Lagrange multiplier $\lambda_i^\star$ for the ith constraint is relatively small, and the constraint is loosened (choosing $u_i > 0$), it is anticipated that the optimal value $p^\star(u, v)$ will not significantly decrease.
- **Outcomes of Tiny Adjustments in Constraints with Small $\nu_i^\star$**:
  - When $\nu_i^\star$ is small and positive, and $v_i > 0$ is chosen, or

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or
  - If $\nu_i^\star$ is large and negative and $v_i > 0$ is selected,

  then in either scenario, the optimal value $p^\star(u, v)$ is expected to increase greatly.
- **Consequences of Loosening a Constraint (Small $\lambda_i^\star$)**:
  If the Lagrange multiplier $\lambda_i^\star$ for the ith constraint is relatively small, and the constraint is loosened (choosing $u_i > 0$), it is anticipated that the optimal value $p^\star(u, v)$ will not significantly decrease.
- **Outcomes of Tiny Adjustments in Constraints with Small $\nu_i^\star$**:
  - When $\nu_i^\star$ is small and positive, and $v_i > 0$ is chosen, or
  - When $\nu_i^\star$ is small and negative, and $v_i < 0$ is opted for,

  in both cases, the optimal value $p^\star(u, v)$ will not significantly decrease.

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or
  - If $\nu_i^\star$ is large and negative and $v_i > 0$ is selected,

  then in either scenario, the optimal value $p^\star(u, v)$ is expected to increase greatly.
- **Consequences of Loosening a Constraint (Small $\lambda_i^\star$)**:
  If the Lagrange multiplier $\lambda_i^\star$ for the ith constraint is relatively small, and the constraint is loosened (choosing $u_i > 0$), it is anticipated that the optimal value $p^\star(u, v)$ will not significantly decrease.
- **Outcomes of Tiny Adjustments in Constraints with Small $\nu_i^\star$**:
  - When $\nu_i^\star$ is small and positive, and $v_i > 0$ is chosen, or
  - When $\nu_i^\star$ is small and negative, and $v_i < 0$ is opted for,

  in both cases, the optimal value $p^\star(u, v)$ will not significantly decrease.

Sensitivity analysis

In scenarios where strong duality holds, we can draw several insights about the sensitivity of optimal solutions in relation to the Lagrange multipliers. These insights are derived from the inequality expressed in equation above:

- **Impact of Tightening a Constraint (Large $\lambda_i^\star$)**:
  When the ith constraint's Lagrange multiplier, $\lambda_i^\star$, holds a substantial value, and if this constraint is tightened (choosing $u_i < 0$), there is a guarantee that the optimal value, denoted by $p^\star(u, v)$, will significantly increase.
- **Effect of Adjusting Constraints with Large Positive or Negative $\nu_i^\star$**:
  - If $\nu_i^\star$ is large and positive and $v_i < 0$ is chosen, or
  - If $\nu_i^\star$ is large and negative and $v_i > 0$ is selected,

  then in either scenario, the optimal value $p^\star(u, v)$ is expected to increase greatly.
- **Consequences of Loosening a Constraint (Small $\lambda_i^\star$)**:
  If the Lagrange multiplier $\lambda_i^\star$ for the ith constraint is relatively small, and the constraint is loosened (choosing $u_i > 0$), it is anticipated that the optimal value $p^\star(u, v)$ will not significantly decrease.
- **Outcomes of Tiny Adjustments in Constraints with Small $\nu_i^\star$**:
  - When $\nu_i^\star$ is small and positive, and $v_i > 0$ is chosen, or
  - When $\nu_i^\star$ is small and negative, and $v_i < 0$ is opted for,

  in both cases, the optimal value $p^\star(u, v)$ will not significantly decrease.

These interpretations provide a framework for understanding how changes in constraints, reflected through their corresponding Lagrange multipliers, impact the optimal solution in problems where strong duality holds.

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u, v)$ along the direction of some i-th basis vector e_i:

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u, v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t \to 0} \frac{p^*(te_i, 0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u, v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t \to 0} \frac{p^*(te_i, 0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

From the inequality Equation 5 and taking the limit $t \to 0$ with $t > 0$ we have

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u, v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t \to 0} \frac{p^*(te_i, 0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

From the inequality Equation 5 and taking the limit $t \to 0$ with $t > 0$ we have

$$\frac{p^*(te_i, 0) - p^*}{t} \geq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \geq -\lambda_i^*$$

Local sensitivity

Suppose now that $p^*(u,v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u,v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t \to 0} \frac{p^*(te_i, 0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

From the inequality Equation 5 and taking the limit $t \to 0$ with $t > 0$ we have

$$\frac{p^*(te_i, 0) - p^*}{t} \geq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \geq -\lambda_i^*$$

For the negative $t < 0$ we have:

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u, v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t \to 0} \frac{p^*(te_i, 0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

From the inequality Equation 5 and taking the limit $t \to 0$ with $t > 0$ we have

$$\frac{p^*(te_i, 0) - p^*}{t} \geq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \geq -\lambda_i^*$$

For the negative $t < 0$ we have:

$$\frac{p^*(te_i, 0) - p^*}{t} \leq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \leq -\lambda_i^*$$

Local sensitivity

Suppose now that $p^*(u,v)$ is differentiable at $u=0, v=0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u,v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t\to 0} \frac{p^*(te_i,0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

From the inequality Equation 5 and taking the limit $t \to 0$ with $t > 0$ we have

$$\frac{p^*(te_i,0) - p^*}{t} \geq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \geq -\lambda_i^*$$

For the negative $t < 0$ we have:

$$\frac{p^*(te_i,0) - p^*}{t} \leq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \leq -\lambda_i^*$$

The same idea can be used to establish the fact about v_i.

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u, v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t \to 0} \frac{p^*(te_i, 0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

From the inequality Equation 5 and taking the limit $t \to 0$ with $t > 0$ we have

$$\frac{p^*(te_i, 0) - p^*}{t} \geq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \geq -\lambda_i^*$$

For the negative $t < 0$ we have:

$$\frac{p^*(te_i, 0) - p^*}{t} \leq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \leq -\lambda_i^*$$

The same idea can be used to establish the fact about v_i. The local sensitivity result Equation 6 provides a way to understand the impact of constraints on the optimal solution x^* of an optimization problem. If a constraint $f_i(x^*)$ is negative at x^*, it's not affecting the optimal solution, meaning small changes to this constraint won't alter the optimal value. In this case, the corresponding optimal Lagrange multiplier will be zero, as per the principle of complementary slackness.

Local sensitivity

Suppose now that $p^*(u, v)$ is differentiable at $u = 0, v = 0$.

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i} \tag{6}$$

To show this result we consider the directional derivative of $p^*(u, v)$ along the direction of some i-th basis vector e_i:

$$\lim_{t \to 0} \frac{p^*(te_i, 0) - p^*(0,0)}{t} = \frac{\partial p^*(0,0)}{\partial u_i}$$

From the inequality Equation 5 and taking the limit $t \to 0$ with $t > 0$ we have

$$\frac{p^*(te_i, 0) - p^*}{t} \geq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \geq -\lambda_i^*$$

For the negative $t < 0$ we have:

$$\frac{p^*(te_i, 0) - p^*}{t} \leq -\lambda_i^* \to \frac{\partial p^*(0,0)}{\partial u_i} \leq -\lambda_i^*$$

The same idea can be used to establish the fact about v_i.
The local sensitivity result Equation 6 provides a way to understand the impact of constraints on the optimal solution x^* of an optimization problem. If a constraint $f_i(x^*)$ is negative at x^*, it's not affecting the optimal solution, meaning small changes to this constraint won't alter the optimal value. In this case, the corresponding optimal Lagrange multiplier will be zero, as per the principle of complementary slackness.
However, if $f_i(x^*) = 0$, meaning the constraint is precisely met at the optimum, then the situation is different. The value of the i-th optimal Lagrange multiplier, λ_i^*, gives us insight into how 'sensitive' or 'active' this constraint is. A small λ_i^* indicates that slight adjustments to the constraint won't significantly affect the optimal value. Conversely, a large λ_i^* implies that even minor changes to the constraint can have a significant impact on the optimal solution.

Mixed Integer Programming

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 &\to \max_{x_1,x_2,x_3,x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 &\leq 14 \\
x_i \in \{0,1\} &\quad \forall i
\end{aligned} \tag{7}
$$

Complexity of MIP

Consider the following Mixed Integer Programming (MIP): Relax it to:

$$
\begin{split}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in \{0, 1\} \quad \forall i
\end{split}
\tag{7}
$$

$$
\begin{split}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in [0, 1] \quad \forall i
\end{split}
\tag{8}
$$

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 &\to \max_{x_1, x_2, x_3, x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 &\leq 14 \\
x_i \in \{0, 1\} \quad &\forall i
\end{aligned}
\tag{7}
$$

Optimal solution

$$
x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.
$$

Relax it to:

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 &\to \max_{x_1, x_2, x_3, x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 &\leq 14 \\
x_i \in [0, 1] \quad &\forall i
\end{aligned}
\tag{8}
$$

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to & \max_{x_1, x_2, x_3, x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 & \\
x_i \in \{0, 1\} \quad \forall i &
\end{aligned} \tag{7}
$$

Optimal solution

$$
x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.
$$

Relax it to:

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to & \max_{x_1, x_2, x_3, x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 & \\
x_i \in [0, 1] \quad \forall i &
\end{aligned} \tag{8}
$$

Optimal solution

$$
x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.
$$

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
& z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
& \text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
& x_i \in \{0,1\} \quad \forall i
\end{aligned}
\tag{7}
$$

Optimal solution

$$x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.$$

Relax it to:

$$
\begin{aligned}
& z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
& \text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
& x_i \in [0,1] \quad \forall i
\end{aligned}
\tag{8}
$$

Optimal solution

$$x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.$$

- Rounding $x_3 = 0$: gives $z = 19$.

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in \{0,1\} \quad \forall i
\end{aligned}
\tag{7}
$$

Optimal solution

$$
x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.
$$

Relax it to:

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in [0,1] \quad \forall i
\end{aligned}
\tag{8}
$$

Optimal solution

$$
x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.
$$

- Rounding $x_3 = 0$: gives $z = 19$.
- Rounding $x_3 = 1$: Infeasible.

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in \{0,1\} \quad \forall i
\end{aligned}
\tag{7}
$$

Optimal solution

$$
x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.
$$

Relax it to:

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in [0,1] \quad \forall i
\end{aligned}
\tag{8}
$$

Optimal solution

$$
x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.
$$

- Rounding $x_3 = 0$: gives $z = 19$.
- Rounding $x_3 = 1$: Infeasible.

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
& z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
& \text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
& x_i \in \{0,1\} \quad \forall i
\end{aligned} \tag{7}
$$

Optimal solution

$$
x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.
$$

Relax it to:

$$
\begin{aligned}
& z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
& \text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
& x_i \in [0,1] \quad \forall i
\end{aligned} \tag{8}
$$

Optimal solution

$$
x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.
$$

- Rounding $x_3 = 0$: gives $z = 19$.
- Rounding $x_3 = 1$: Infeasible.

! MIP is much harder, than LP

- Naive rounding of LP relaxation of the initial MIP problem might lead to infeasible or suboptimal solution.

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
& z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4} \\
& \text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
& x_i \in \{0, 1\} \quad \forall i
\end{aligned}
\tag{7}
$$

Optimal solution

$$
x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.
$$

Relax it to:

$$
\begin{aligned}
& z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4} \\
& \text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
& x_i \in [0, 1] \quad \forall i
\end{aligned}
\tag{8}
$$

Optimal solution

$$
x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.
$$

- Rounding $x_3 = 0$: gives $z = 19$.
- Rounding $x_3 = 1$: Infeasible.

❗ MIP is much harder, than LP

- Naive rounding of LP relaxation of the initial MIP problem might lead to infeasible or suboptimal solution.
- General MIP is NP-hard.

Complexity of MIP

Consider the following Mixed Integer Programming (MIP):

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in \{0,1\} \quad \forall i
\end{aligned} \tag{7}
$$

Optimal solution

$$
x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ and } z = 21.
$$

Relax it to:

$$
\begin{aligned}
z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1,x_2,x_3,x_4} \\
\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
x_i \in [0,1] \quad \forall i
\end{aligned} \tag{8}
$$

Optimal solution

$$
x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.
$$

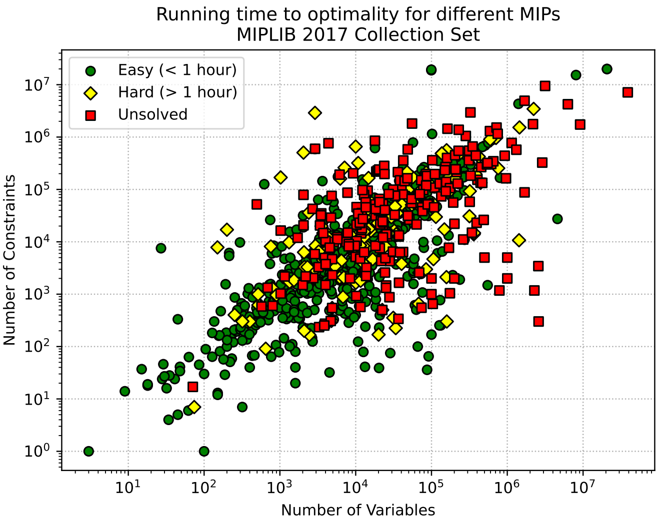
- Rounding $x_3 = 0$: gives $z = 19$.
- Rounding $x_3 = 1$: Infeasible.

❗ MIP is much harder, than LP

- Naive rounding of LP relaxation of the initial MIP problem might lead to infeasible or suboptimal solution.
- General MIP is NP-hard.
- However, if the coefficient matrix of an MIP is a *totally unimodular matrix*, then it can be solved in polynomial time.

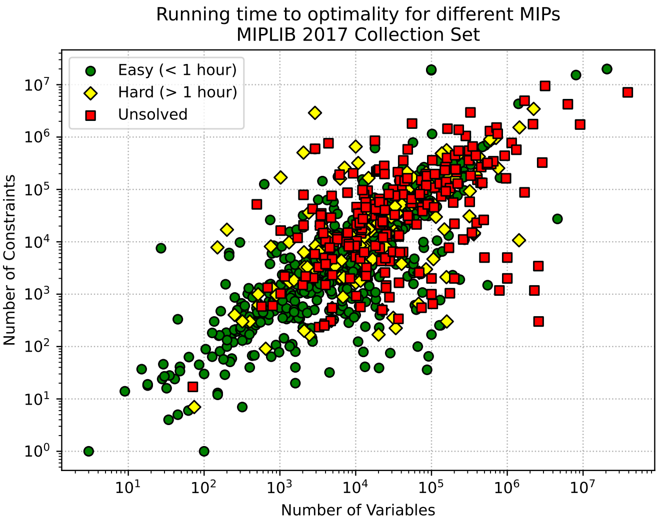
Unpredictable complexity of MIP

- It is hard to predict what will be solved quickly and what will take a long time



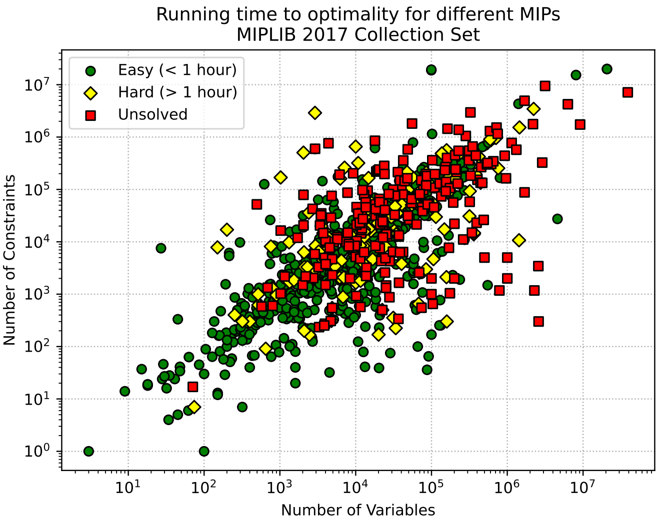
Unpredictable complexity of MIP

- It is hard to predict what will be solved quickly and what will take a long time
- Dataset



Unpredictable complexity of MIP

- It is hard to predict what will be solved quickly and what will take a long time
- Dataset
- Source code



Hardware progress vs Software progress

What would you choose, assuming, that the question posed correctly (you can compile software for any hardware and the problem is the same for both options)? We will consider the time period from 1992 to 2023.

Hardware

Solving MIP with an old software on the modern hardware

Software

Solving MIP with a modern software on the old hardware

Hardware progress vs Software progress

What would you choose, assuming, that the question posed correctly (you can compile software for any hardware and the problem is the same for both options)? We will consider the time period from 1992 to 2023.

Hardware

Solving MIP with an old software on the modern hardware

$\approx 1.664.510$ x speedup

Moore's law states, that computational power doubles every 18 monthes.

Software

Solving MIP with a modern software on the old hardware

$\approx 2.349.000$ x speedup

R. Bixby conducted an intensive experiment with benchmarking all CPLEX software version starting from 1992 to 2007 and measured overall software progress (29000 times), later (in 2009) he was a cofounder of Gurobi optimization software, which gives additional ≈ 81 speedup on MILP.

[1] R. Bixby report Recent study

Hardware progress vs Software progress

What would you choose, assuming, that the question posed correctly (you can compile software for any hardware and the problem is the same for both options)? We will consider the time period from 1992 to 2023.

Hardware

Solving MIP with an old software on the modern hardware

$\approx 1.664.510$ x speedup

Moore's law states, that computational power doubles every 18 monthes.

Software

Solving MIP with a modern software on the old hardware

$\approx 2.349.000$ x speedup

R. Bixby conducted an intensive experiment with benchmarking all CPLEX software version starting from 1992 to 2007 and measured overall software progress (29000 times), later (in 2009) he was a cofounder of Gurobi optimization software, which gives additional ≈ 81 speedup on MILP.

It turns out that if you need to solve a MILP, it is better to use an old computer and modern methods than vice versa, the newest computer and methods of the early 1990s![1]

[1] R. Bixby report Recent study

Sources

- Optimization Theory (MATH4230) course @ CUHK by Professor Tieyong Zeng