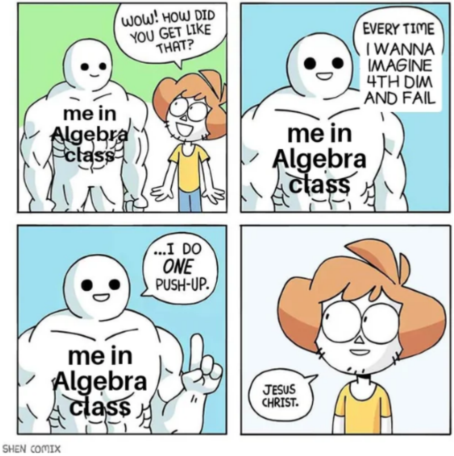


Introduction to Optimization

K. R. Sahasranand

Data Science

sahasranand@iitpkd.ac.in

Surprise online test tomorrow at 6:00 am

Surprise online test tomorrow at 6:00 am

Surprise online test tomorrow at 6:00 am

- Syllabus – modules I and II
- Time to prepare ~ 20 hours

Surprise online test tomorrow at 6:00 am

- Syllabus – modules I and II
- Time to prepare ~ 20 hours
- Will have to drink 3 cups of coffee **every hour** for module I

Surprise online test tomorrow at 6:00 am

- Syllabus – modules I and II
- Time to prepare ~ 20 hours
- Will have to drink 3 cups of coffee **every hour** for module I
- Will have to drink 4 cups of coffee **every hour** for module II

Surprise online test tomorrow at 6:00 am

- Syllabus – modules I and II
- Time to prepare ~ 20 hours
- Will have to drink 3 cups of coffee **every hour** for module I
- Will have to drink 4 cups of coffee **every hour** for module II

Nobody:

Mom: *"Don't drink too much coffee...*

Surprise online test tomorrow at 6:00 am

- Syllabus – modules I and II
- Time to prepare ~ 20 hours
- Will have to drink 3 cups of coffee **every hour** for module I
- Will have to drink 4 cups of coffee **every hour** for module II

Nobody:

Mom: *"Don't drink too much coffee... But, since this test counts towards your CGPA, you may drink up to 72 cups."*

Surprise online test tomorrow at 6:00 am

- Syllabus – modules I and II
- Time to prepare ~ 20 hours
- Will have to drink 3 cups of coffee **every hour** for module I
- Will have to drink 4 cups of coffee **every hour** for module II

Nobody:

Mom: *"Don't drink too much coffee... But, since this test counts towards your CGPA, you may drink up to 72 cups."*

- You score 4 points for every hour spent on module I and 5 points for every hour spent on module II

Surprise online test tomorrow at 6:00 am

- Syllabus – modules I and II
- Time to prepare ~ 20 hours
- Will have to drink 3 cups of coffee **every hour** for module I
- Will have to drink 4 cups of coffee **every hour** for module II

Nobody:

Mom: *"Don't drink too much coffee... But, since this test counts towards your CGPA, you may drink up to 72 cups."*

- You score 4 points for every hour spent on module I and 5 points for every hour spent on module II

How many hours for each module?

Another example

$$
\begin{aligned}
\text{maximize}\quad & x_1 + x_2 \\
\text{subject to}\quad & x_2 - x_1 \leq 1 \\
& x_1 + 6x_2 \leq 15 \\
& 4x_1 - x_2 \leq 10 \\
& x_1 \geq 0,\ x_2 \geq 0.
\end{aligned}
$$

(continue on board)

Another example

$$\begin{aligned} \text{maximize } & x_1 + x_2 \\ \text{subject to } & x_2 - x_1 \le 1 \\ & x_1 + 6x_2 \le 15 \\ & 4x_1 - x_2 \le 10 \\ & x_1 \ge 0, \ x_2 \ge 0. \end{aligned}$$

(continue on board)

To-do:

- show vector c and feasible region

Another example

$$
\begin{aligned}
\text{maximize } & x_1 + x_2 \\
\text{subject to } & x_2 - x_1 \leq 1 \\
& x_1 + 6x_2 \leq 15 \\
& 4x_1 - x_2 \leq 10 \\
& x_1 \geq 0,\ x_2 \geq 0.
\end{aligned}
$$

(continue on board)

To-do:

- show vector c and feasible region *unique solution*

Another example

$$
\begin{aligned}
\text{maximize } & x_1 + x_2 \\
\text{subject to } & x_2 - x_1 \le 1 \\
& x_1 + 6x_2 \le 15 \\
& 4x_1 - x_2 \le 10 \\
& x_1 \ge 0,\ x_2 \ge 0.
\end{aligned}
$$

(continue on board)

To-do:

- show vector c and feasible region *unique solution*
- replace $c = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ with $\begin{bmatrix} \frac{1}{6} & 1 \end{bmatrix}^\top$

Another example

$$\begin{aligned} \text{maximize } & x_1 + x_2 \\ \text{subject to } & x_2 - x_1 \leq 1 \\ & x_1 + 6x_2 \leq 15 \\ & 4x_1 - x_2 \leq 10 \\ & x_1 \geq 0,\ x_2 \geq 0. \end{aligned}$$

(continue on board)

To-do:

- show vector c and feasible region *unique solution*
- replace $c = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ with $\begin{bmatrix} \frac{1}{6} & 1 \end{bmatrix}^\top$ *infinitely many solutions*

Another example

$$\begin{aligned} \text{maximize } & x_1 + x_2 \\ \text{subject to } & x_2 - x_1 \le 1 \\ & x_1 + 6x_2 \le 15 \\ & 4x_1 - x_2 \le 10 \\ & x_1 \ge 0,\ x_2 \ge 0. \end{aligned}$$

(continue on board)

To-do:

- show vector c and feasible region *unique solution*
- replace $c = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ with $\begin{bmatrix} \frac{1}{6} & 1 \end{bmatrix}^\top$ *infinitely many solutions*
- reverse 1^{st} and 3^{rd}

Another example

$$\begin{aligned} \text{maximize } & x_1 + x_2 \\ \text{subject to } & x_2 - x_1 \leq 1 \\ & x_1 + 6x_2 \leq 15 \\ & 4x_1 - x_2 \leq 10 \\ & x_1 \geq 0,\ x_2 \geq 0. \end{aligned}$$

(continue on board)

To-do:

- show vector c and feasible region — *unique solution*
- replace $c = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ with $\begin{bmatrix} \frac{1}{6} & 1 \end{bmatrix}^\top$ — *infinitely many solutions*
- reverse 1^{st} and 3^{rd} — *infeasible*

Another example

$$\begin{aligned} \text{maximize } & x_1 + x_2 \\ \text{subject to } & x_2 - x_1 \leq 1 \\ & x_1 + 6x_2 \leq 15 \\ & 4x_1 - x_2 \leq 10 \\ & x_1 \geq 0, \; x_2 \geq 0. \end{aligned}$$

(continue on board)

To-do:

- show vector c and feasible region — *unique solution*
- replace $c = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ with $\begin{bmatrix} \frac{1}{6} & 1 \end{bmatrix}^\top$ — *infinitely many solutions*
- reverse 1^{st} and 3^{rd} — *infeasible*
- remove 2^{nd} and 3^{rd}

Another example

$$\begin{aligned} \text{maximize } & x_1 + x_2 \\ \text{subject to } & x_2 - x_1 \leq 1 \\ & x_1 + 6x_2 \leq 15 \\ & 4x_1 - x_2 \leq 10 \\ & x_1 \geq 0, \; x_2 \geq 0. \end{aligned}$$

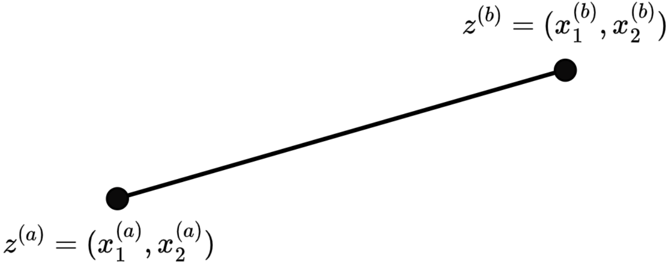
(continue on board)

To-do:

- show vector c and feasible region *unique solution*
- replace $c = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ with $\begin{bmatrix} \frac{1}{6} & 1 \end{bmatrix}^\top$ *infinitely many solutions*
- reverse 1^{st} and 3^{rd} *infeasible*
- remove 2^{nd} and 3^{rd} *unbounded*

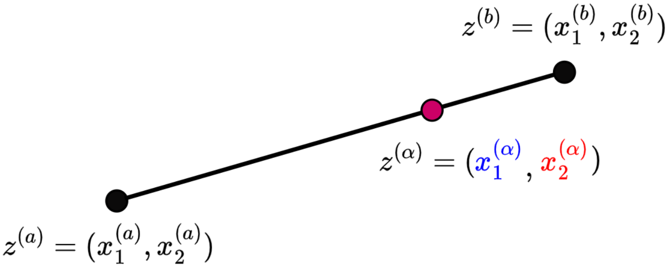
Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



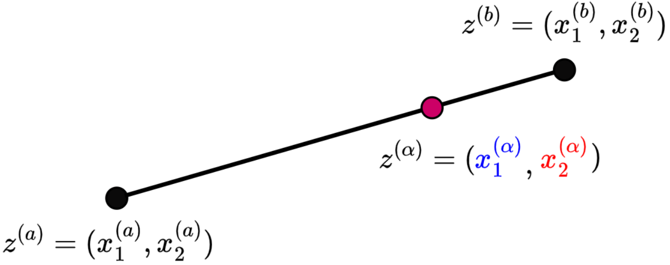
Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



Why it suffices to look at *corner points*?

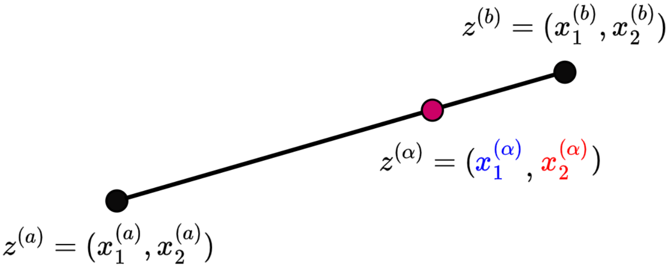
$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



$$z^{(\alpha)} = \alpha z^{(a)} + (1-\alpha) z^{(b)} \quad ; \; 0 \le \alpha \le 1.$$

Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



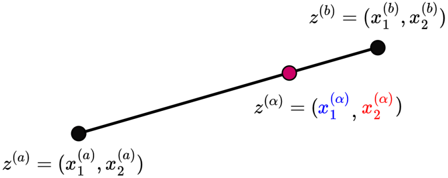
$$z^{(\alpha)} = \alpha z^{(a)} + (1-\alpha) z^{(b)} \quad ; \; 0 \leq \alpha \leq 1.$$

That is,

$$x_1^{(\alpha)} = \alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}$$
$$x_2^{(\alpha)} = \alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}$$

Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



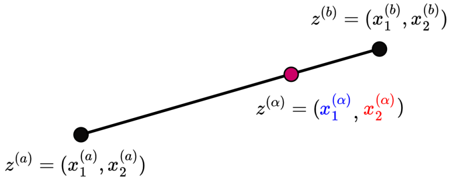
$$x_1^{(\alpha)} = \alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}$$
$$x_2^{(\alpha)} = \alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}$$
$$;\ 0 \le \alpha \le 1.$$

Objective function value at $z^{(\alpha)}$:

$$f(z^{(\alpha)}) = c^\top z^{(\alpha)}$$

Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



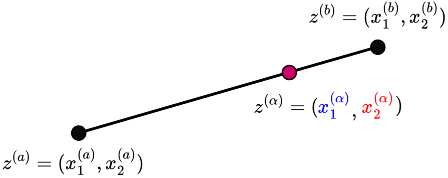
$$x_1^{(\alpha)} = \alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}$$
$$x_2^{(\alpha)} = \alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}$$
$$;\ 0 \le \alpha \le 1.$$

Objective function value at $z^{(\alpha)}$:

$$\begin{aligned} f(z^{(\alpha)}) &= c^\top z^{(\alpha)} \\ &= c_1 x_1^{(\alpha)} + c_2 x_2^{(\alpha)} \end{aligned}$$

Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



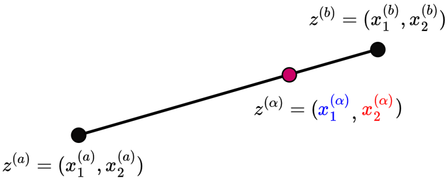
$$x_1^{(\alpha)} = \alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}$$
$$x_2^{(\alpha)} = \alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}$$
$$;\ 0 \leq \alpha \leq 1.$$

Objective function value at $z^{(\alpha)}$:

$$\begin{aligned} f(z^{(\alpha)}) &= c^\top z^{(\alpha)} \\ &= c_1 x_1^{(\alpha)} + c_2 x_2^{(\alpha)} \\ &= c_1 \left(\alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}\right) + c_2 \left(\alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}\right) \end{aligned}$$

Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



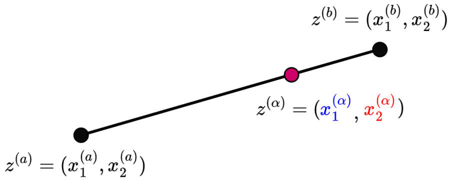
$$x_1^{(\alpha)} = \alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}$$
$$x_2^{(\alpha)} = \alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}$$
$$;\ 0 \leq \alpha \leq 1.$$

Objective function value at $z^{(\alpha)}$:

$$\begin{aligned} f(z^{(\alpha)}) &= c^\top z^{(\alpha)} \\ &= c_1 x_1^{(\alpha)} + c_2 x_2^{(\alpha)} \\ &= c_1 \left(\alpha x_1^{(a)} + (1-\alpha) x_1^{(b)} \right) + c_2 \left(\alpha x_2^{(a)} + (1-\alpha) x_2^{(b)} \right) \\ &= \alpha \left(c_1 x_1^{(a)} + c_2 x_2^{(a)} \right) + (1-\alpha) \left(c_1 x_1^{(b)} + c_2 x_2^{(b)} \right) \end{aligned}$$

Why it suffices to look at *corner points*?

$\max f(z)$ subject to some linear constraints where $f(z) = c^\top z$



$$x_1^{(\alpha)} = \alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}$$
$$x_2^{(\alpha)} = \alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}$$
$$;\ 0 \le \alpha \le 1.$$

Objective function value at $z^{(\alpha)}$:

$$\begin{aligned}
f(z^{(\alpha)}) &= c^\top z^{(\alpha)} \\
&= c_1 x_1^{(\alpha)} + c_2 x_2^{(\alpha)} \\
&= c_1 \left(\alpha x_1^{(a)} + (1-\alpha) x_1^{(b)}\right) + c_2 \left(\alpha x_2^{(a)} + (1-\alpha) x_2^{(b)}\right) \\
&= \alpha \left(c_1 x_1^{(a)} + c_2 x_2^{(a)}\right) + (1-\alpha) \left(c_1 x_1^{(b)} + c_2 x_2^{(b)}\right) \\
&= \alpha f(z^{(a)}) + (1-\alpha) f(z^{(b)})
\end{aligned}$$

Top 10 algorithms from the 20^{th} century

Computing in Science & Engineering, American Institute of Physics and the IEEE Computer Society, Jan 2000.

Simplex algorithm – bird's eye view

Algorithms, Dasgupta-Papadimitriou-Vazirani [online]

```
Let v be any vertex of the feasible region.
while ∃ a neighbor v' of v with better objective value:
        set v = v'
```

Simplex algorithm – bird's eye view

Algorithms, Dasgupta-Papadimitriou-Vazirani [online]

```
Let v be any vertex of the feasible region.
while ∃ a neighbor v' of v with better objective value:
        set v = v'
```

On each iteration, the algorithm has **two** tasks:

Simplex algorithm – bird's eye view

Algorithms, Dasgupta-Papadimitriou-Vazirani [online]

```
Let v be any vertex of the feasible region.
while ∃ a neighbor v' of v with better objective value:
        set v = v'
```

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).

Simplex algorithm – bird's eye view

Algorithms, Dasgupta-Papadimitriou-Vazirani [online]

`Let` v `be any vertex of the feasible region.`

`while` $\exists$ `a neighbor` v' `of` v `with better objective value:`

`set` $v = v'$

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Simplex algorithm – bird's eye view

Algorithms, Dasgupta-Papadimitriou-Vazirani [online]

```
Let v be any vertex of the feasible region.
while ∃ a neighbor v' of v with better objective value:
        set v = v'
```

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Both tasks are easy if the vertex is at the origin!

Simplex algorithm – bird's eye view

Algorithms, Dasgupta-Papadimitriou-Vazirani [online]

```
Let v be any vertex of the feasible region.
while ∃ a neighbor v' of v with better objective value:
        set v = v'
```

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Both tasks are easy if the vertex is at the origin!

But what if our current vertex u is elsewhere?

Simplex algorithm – bird's eye view

Algorithms, Dasgupta-Papadimitriou-Vazirani [online]

`Let` v `be any vertex of the feasible region.`

`while` $\exists$ `a neighbor` v' `of` v `with better objective value:`

    `set` $v = v'$

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Both tasks are easy if the vertex is at the origin!

But what if our current vertex u is elsewhere?

Transform u *into the origin, by shifting the coordinate system from the usual* $(x_1, x_2, \cdots, x_n)$ *to the "local view" from* u*.*

The two tasks at the origin

"Both tasks are easy if the vertex is at the origin!"

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Consider the LP:

$$\text{maximize } c^\top x \text{ subject to } Ax \geq b, x \geq 0 \quad ; \; x \in \mathbb{R}^n.$$

The two tasks at the origin

"Both tasks are easy if the vertex is at the origin!"

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Consider the LP:

$$\text{maximize } c^\top x \text{ subject to } Ax \geq b, x \geq 0 \quad ; \; x \in \mathbb{R}^n.$$

Origin is feasible $\implies$ Origin is a vertex

since it is the unique point at which the n inequalities

$x_1 \geq 0, \; x_2 \geq 0, \; \cdots, \; x_n \geq 0$ are tight.

The two tasks at the origin

"Both tasks are easy if the vertex is at the origin!"

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Consider the LP:

$$\text{maximize } c^\top x \text{ subject to } Ax \geq b, x \geq 0 \quad ; \; x \in \mathbb{R}^n.$$

Origin is feasible $\implies$ Origin is a vertex

since it is the unique point at which the n inequalities

$x_1 \geq 0, \; x_2 \geq 0, \; \cdots, \; x_n \geq 0$ are tight.

The origin is optimal if and only if all $c_i \leq 0$.

The two tasks at the origin

"Both tasks are easy if the vertex is at the origin!"

On each iteration, the algorithm has **two** tasks:

1. Check whether the current vertex is optimal (and if so, halt).
2. Determine where to move next.

Consider the LP:

maximize $c^\top x$ subject to $Ax \geq b, x \geq 0 \quad ; \; x \in \mathbb{R}^n$.

Origin is feasible $\implies$ Origin is a vertex

since it is the unique point at which the n inequalities

$x_1 \geq 0, \; x_2 \geq 0, \; \cdots, \; x_n \geq 0$ are tight.

The origin is optimal if and only if all $c_i \leq 0$.

Increase some x_i for which $c_i > 0$.

Increase by how much? Until we hit some other constraint.

Solve using simplex algorithm

"*always operate at the origin*"

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \le 4 \quad (1)$$

$$x_1 + 2x_2 \le 9 \quad (2)$$

$$-x_1 + x_2 \le 3 \quad (3)$$

$$x_1 \ge 0 \quad (4)$$

$$x_2 \ge 0 \quad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(5) **Objective value**:

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \leq 4 \qquad (1)$$

$$x_1 + 2x_2 \leq 9 \qquad (2)$$

$$-x_1 + x_2 \leq 3 \qquad (3)$$

$$x_1 \geq 0 \qquad (4)$$

$$x_2 \geq 0 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(5) **Objective value**: 0

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \leq 4 \qquad (1)$$

$$x_1 + 2x_2 \leq 9 \qquad (2)$$

$$-x_1 + x_2 \leq 3 \qquad (3)$$

$$x_1 \geq 0 \qquad (4)$$

$$x_2 \geq 0 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(5)

Objective value: 0

Move:

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \leq 4 \qquad (1)$$

$$x_1 + 2x_2 \leq 9 \qquad (2)$$

$$-x_1 + x_2 \leq 3 \qquad (3)$$

$$x_1 \geq 0 \qquad (4)$$

$$x_2 \geq 0 \qquad (5)$$

Solve using simplex algorithm

*"always operate at the **origin**"*

Current vertex: (4),(5) **Objective value**: 0

Move: increase x_2

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$\begin{aligned}
2x_1 - x_2 &\leq 4 && (1)\\
x_1 + 2x_2 &\leq 9 && (2)\\
-x_1 + x_2 &\leq 3 && (3)\\
x_1 &\geq 0 && (4)\\
x_2 &\geq 0 && (5)
\end{aligned}$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(5) **Objective value**: 0

Move: increase x_2

(5) is released, (3) is tightened

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \leq 4 \qquad (1)$$

$$x_1 + 2x_2 \leq 9 \qquad (2)$$

$$-x_1 + x_2 \leq 3 \qquad (3)$$

$$x_1 \geq 0 \qquad (4)$$

$$x_2 \geq 0 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(5)

Objective value: 0

Move: increase x_2

STOP at $x_2 = ?$

(5) is released, (3) is tightened

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \leq 4 \qquad (1)$$

$$x_1 + 2x_2 \leq 9 \qquad (2)$$

$$-x_1 + x_2 \leq 3 \qquad (3)$$

$$x_1 \geq 0 \qquad (4)$$

$$x_2 \geq 0 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(5) **Objective value**: 0

Move: increase x_2

STOP at $x_2 = 3$

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \le 4 \qquad (1)$$

$$x_1 + 2x_2 \le 9 \qquad (2)$$

$$-x_1 + x_2 \le 3 \qquad (3)$$

$$x_1 \ge 0 \qquad (4)$$

$$x_2 \ge 0 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(5)

Objective value: 0

Move: increase x_2

STOP at $x_2 = 3$

$\boxed{\textbf{New vertex: } (4),(3)}$

$$\max\ 2x_1 + 5x_2 \text{ subject to}$$

$$2x_1 - x_2 \leq 4 \qquad (1)$$

$$x_1 + 2x_2 \leq 9 \qquad (2)$$

$$-x_1 + x_2 \leq 3 \qquad (3)$$

$$x_1 \geq 0 \qquad (4)$$

$$x_2 \geq 0 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the **origin***"

Current vertex: (4),(5)

Objective value: 0

Move: increase x_2

STOP at $x_2 = 3$

$\boxed{\textbf{New vertex: } (4),(3)}$

$$y_1 = x_1$$
$$y_2 = 3 + x_1 - x_2$$

$\max\ 2x_1 + 5x_2$ subject to

$2x_1 - x_2 \leq 4$	(1)
$x_1 + 2x_2 \leq 9$	(2)
$-x_1 + x_2 \leq 3$	(3)
$x_1 \geq 0$	(4)
$x_2 \geq 0$	(5)

Solve using simplex algorithm

"*always operate at the **origin***"

Current vertex: (4),(3) **Objective value**:

max $15 + 7y_1 - 5y_2$ subject to

$$y_1 + y_2 \leq 7 \quad (1)$$

$$3y_1 - 2y_2 \leq 3 \quad (2)$$

$$y_2 \geq 0 \quad (3)$$

$$y_1 \geq 0 \quad (4)$$

$$-y_1 + y_2 \leq 3 \quad (5)$$

Solve using simplex algorithm

"*always operate at the **origin***"

Current vertex: (4),(3) **Objective value**: 15

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \leq 7 \qquad (1)$$

$$3y_1 - 2y_2 \leq 3 \qquad (2)$$

$$y_2 \geq 0 \qquad (3)$$

$$y_1 \geq 0 \qquad (4)$$

$$-y_1 + y_2 \leq 3 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(3)

Objective value: 15

Move:

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \leq 7 \qquad (1)$$

$$3y_1 - 2y_2 \leq 3 \qquad (2)$$

$$y_2 \geq 0 \qquad (3)$$

$$y_1 \geq 0 \qquad (4)$$

$$-y_1 + y_2 \leq 3 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the **origin***"

Current vertex: (4),(3) **Objective value**: 15

Move: increase y_1

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \le 7 \qquad (1)$$

$$3y_1 - 2y_2 \le 3 \qquad (2)$$

$$y_2 \ge 0 \qquad (3)$$

$$y_1 \ge 0 \qquad (4)$$

$$-y_1 + y_2 \le 3 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(3)

Objective value: 15

Move: increase y_1

$\boxed{\text{(4) is released, (2) is tightened}}$

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \leq 7 \qquad (1)$$

$$3y_1 - 2y_2 \leq 3 \qquad (2)$$

$$y_2 \geq 0 \qquad (3)$$

$$y_1 \geq 0 \qquad (4)$$

$$-y_1 + y_2 \leq 3 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(3)

Objective value: 15

Move: increase y_1

STOP at $y_1 = ?$

$\boxed{\text{(4) is released, (2) is tightened}}$

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \leq 7 \qquad (1)$$

$$3y_1 - 2y_2 \leq 3 \qquad (2)$$

$$y_2 \geq 0 \qquad (3)$$

$$y_1 \geq 0 \qquad (4)$$

$$-y_1 + y_2 \leq 3 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(3) **Objective value**: 15

Move: increase y_1

STOP at $y_1 = 1$

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \leq 7 \quad (1)$$

$$3y_1 - 2y_2 \leq 3 \quad (2)$$

$$y_2 \geq 0 \quad (3)$$

$$y_1 \geq 0 \quad (4)$$

$$-y_1 + y_2 \leq 3 \quad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(3)

Objective value: 15

Move: increase y_1

STOP at $y_1 = 1$

$\boxed{\textbf{New vertex}\text{: (2),(3)}}$

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \leq 7 \qquad (1)$$

$$3y_1 - 2y_2 \leq 3 \qquad (2)$$

$$y_2 \geq 0 \qquad (3)$$

$$y_1 \geq 0 \qquad (4)$$

$$-y_1 + y_2 \leq 3 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (4),(3)

Objective value: 15

Move: increase y_1

STOP at $y_1 = 1$

New vertex: (2),(3)

$$z_1 = 3 - 3y_1 + 2y_2$$
$$z_2 = y_2$$

$$\max\ 15 + 7y_1 - 5y_2 \text{ subject to}$$

$$y_1 + y_2 \leq 7 \qquad (1)$$
$$3y_1 - 2y_2 \leq 3 \qquad (2)$$
$$y_2 \geq 0 \qquad (3)$$
$$y_1 \geq 0 \qquad (4)$$
$$-y_1 + y_2 \leq 3 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (2),(3)

Objective value:

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \text{ subject to}$$

$$-\frac{1}{3}z_1 + \frac{5}{3}z_2 \le 6 \qquad (1)$$

$$z_1 \ge 0 \qquad (2)$$

$$z_2 \ge 0 \qquad (3)$$

$$\frac{1}{3}z_1 - \frac{2}{3}z_2 \le 1 \qquad (4)$$

$$\frac{1}{3}z_1 + \frac{1}{3}z_2 \le 4 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the* ***origin***"

Current vertex: (2),(3) **Objective value:** 22

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \text{ subject to}$$

$$-\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6 \qquad (1)$$

$$z_1 \geq 0 \qquad (2)$$

$$z_2 \geq 0 \qquad (3)$$

$$\frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1 \qquad (4)$$

$$\frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (2),(3) **Objective value**: 22

Move:

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \text{ subject to}$$

$$-\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6 \qquad (1)$$

$$z_1 \geq 0 \qquad (2)$$

$$z_2 \geq 0 \qquad (3)$$

$$\frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1 \qquad (4)$$

$$\frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex:

Objective value: 22

Move:

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \text{ subject to}$$

$$-\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6 \qquad (1)$$

$$z_1 \geq 0 \qquad (2)$$

$$z_2 \geq 0 \qquad (3)$$

$$\frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1 \qquad (4)$$

$$\frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the* ***origin***"

Current vertex: (2),(3) **Objective value**: 22

Move: origin is optimal!
(all $c_i < 0$)

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \text{ subject to}$$

$$-\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6 \qquad (1)$$

$$z_1 \geq 0 \qquad (2)$$

$$z_2 \geq 0 \qquad (3)$$

$$\frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1 \qquad (4)$$

$$\frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (2),(3) **Objective value**: $\boxed{22}$

Move: origin is optimal!
(all $c_i < 0$)

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \text{ subject to}$$

$$-\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6 \quad (1)$$

$$z_1 \geq 0 \quad (2)$$

$$z_2 \geq 0 \quad (3)$$

$$\frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1 \quad (4)$$

$$\frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4 \quad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (2),(3)

Objective value: $\boxed{22}$

Move: origin is optimal!
(all $c_i < 0$)

Solve (2),(3) in the original LP:

$$\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \text{ subject to}$$

$$-\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6 \qquad (1)$$

$$z_1 \geq 0 \qquad (2)$$

$$z_2 \geq 0 \qquad (3)$$

$$\frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1 \qquad (4)$$

$$\frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4 \qquad (5)$$

Solve using simplex algorithm

"*always operate at the origin*"

Current vertex: (2),(3)

Objective value: $\boxed{22}$

Move: origin is optimal!
(all $c_i < 0$)

Solve (2),(3) in the original LP:

$$\boxed{(x_1, x_2) = (1, 4)}$$

$\max 22 - \frac{7}{3} z_1 - \frac{1}{3} z_2$ subject to

$$-\frac{1}{3} z_1 + \frac{5}{3} z_2 \leq 6 \qquad (1)$$

$$z_1 \geq 0 \qquad (2)$$

$$z_2 \geq 0 \qquad (3)$$

$$\frac{1}{3} z_1 - \frac{2}{3} z_2 \leq 1 \qquad (4)$$

$$\frac{1}{3} z_1 + \frac{1}{3} z_2 \leq 4 \qquad (5)$$

Back to the first example

solve using simplex

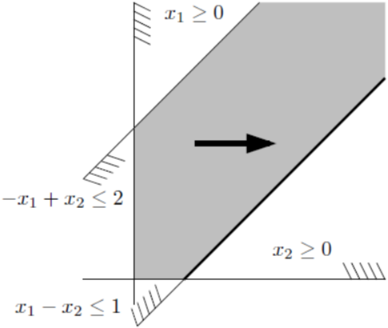
$$\begin{aligned} \text{maximize } & 4x_1 + 5x_2 \\ \text{subject to } & x_1 + x_2 \leq 20 \\ & 3x_1 + 4x_2 \leq 72 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

(complete on board)

Another interesting example

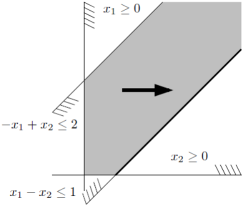
Unbounded LP

$$\begin{aligned} &\text{maximize } x_1 \\ &\text{subject to } x_1 - x_2 \le 1 && (1) \\ &\quad -x_1 + x_2 \le 2 && (2) \\ &\quad x_1 \ge 0 && (3) \\ &\quad x_2 \ge 0 && (4) \end{aligned}$$



Another interesting example

Unbounded LP



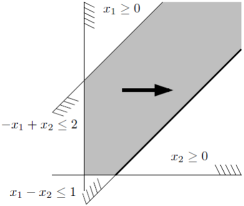
$$
\begin{array}{lll}
\text{maximize } x_1 & & \\
\text{subject to } x_1 - x_2 \leq 1 & & (1) \\
-x_1 + x_2 \leq 2 & & (2) \\
x_1 \geq 0 & & (3) \\
x_2 \geq 0 & & (4)
\end{array}
$$

Current vertex: (3),(4)

Objective value: 0

Another interesting example

Unbounded LP



$$\begin{aligned} \text{maximize } & x_1 & \\ \text{subject to } & x_1 - x_2 \leq 1 & (1) \\ & -x_1 + x_2 \leq 2 & (2) \\ & x_1 \geq 0 & (3) \\ & x_2 \geq 0 & (4) \end{aligned}$$

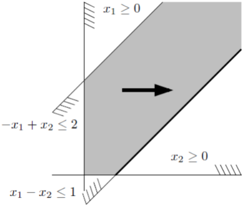
Current vertex: (3),(4)

Objective value: 0

Move: increase x_1

Another interesting example

Unbounded LP



$$
\begin{aligned}
\text{maximize } & x_1 & \\
\text{subject to } & x_1 - x_2 \leq 1 & (1)\\
& -x_1 + x_2 \leq 2 & (2)\\
& x_1 \geq 0 & (3)\\
& x_2 \geq 0 & (4)
\end{aligned}
$$

Current vertex: (3),(4)

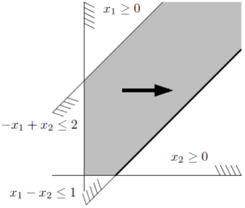
Objective value: 0

Move: increase x_1

STOP at $x_1 = 1$

Another interesting example

Unbounded LP



maximize x_1

subject to $x_1 - x_2 \leq 1$ (1)

$-x_1 + x_2 \leq 2$ (2)

$x_1 \geq 0$ (3)

$x_2 \geq 0$ (4)

Current vertex: (3),(4)

Objective value: 0

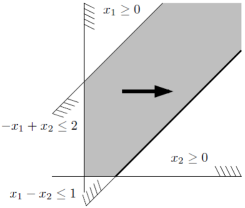
Move: increase x_1

STOP at $x_1 = 1$

New vertex: (1),(4)

Another interesting example

Unbounded LP



$$
\begin{aligned}
\text{maximize } & x_1 \\
\text{subject to } & x_1 - x_2 \leq 1 && (1) \\
& -x_1 + x_2 \leq 2 && (2) \\
& x_1 \geq 0 && (3) \\
& x_2 \geq 0 && (4)
\end{aligned}
$$

Current vertex: (3),(4)

Objective value: 0

Move: increase x_1

STOP at $x_1 = 1$

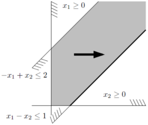
New vertex: (1),(4)

$$y_2 = x_2$$

$$y_1 = 1 - x_1 + x_2$$

Another interesting example

Unbounded LP



maximize x_1

subject to $x_1 - x_2 \leq 1$ (1)

$-x_1 + x_2 \leq 2$ (2)

$x_1 \geq 0$ (3)

$x_2 \geq 0$ (4)

Current vertex: (3),(4)

Objective value: 0

Move: increase x_1

STOP at $x_1 = 1$

New vertex: (1),(4)

$y_2 = x_2$

$y_1 = 1 - x_1 + x_2$

---

maximize $1 - y_1 + y_2$

subject to $y_1 \geq 0$ (1)

$y_1 \leq 3$ (2)

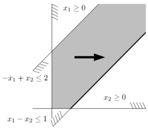
$y_1 - y_2 \leq 1$ (3)

$y_2 \geq 0$ (4)

Current vertex: (1),(4)

Another interesting example

Unbounded LP



maximize x_1

subject to $x_1 - x_2 \leq 1$ (1)

$-x_1 + x_2 \leq 2$ (2)

$x_1 \geq 0$ (3)

$x_2 \geq 0$ (4)

Current vertex: (3),(4)

Objective value: 0

Move: increase x_1

STOP at $x_1 = 1$

New vertex: (1),(4)

$y_2 = x_2$

$y_1 = 1 - x_1 + x_2$

---

maximize $1 - y_1 + y_2$

subject to $y_1 \geq 0$ (1)

$y_1 \leq 3$ (2)

$y_1 - y_2 \leq 1$ (3)

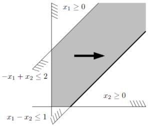
$y_2 \geq 0$ (4)

Current vertex: (1),(4)

Objective value: 1

Another interesting example

Unbounded LP



maximize x_1

subject to

$$x_1 - x_2 \leq 1 \quad (1)$$
$$-x_1 + x_2 \leq 2 \quad (2)$$
$$x_1 \geq 0 \quad (3)$$
$$x_2 \geq 0 \quad (4)$$

Current vertex: (3),(4)

Objective value: 0

Move: increase x_1

STOP at $x_1 = 1$

New vertex: (1),(4)

$$y_2 = x_2$$
$$y_1 = 1 - x_1 + x_2$$

---

maximize $1 - y_1 + y_2$

subject to

$$y_1 \geq 0 \quad (1)$$
$$y_1 \leq 3 \quad (2)$$
$$y_1 - y_2 \leq 1 \quad (3)$$
$$y_2 \geq 0 \quad (4)$$

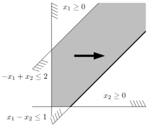
Current vertex: (1),(4)

Objective value: 1

Move: increase y_2

Another interesting example

Unbounded LP



$$
\begin{aligned}
\text{maximize } & x_1 & \\
\text{subject to } & x_1 - x_2 \leq 1 & (1)\\
& -x_1 + x_2 \leq 2 & (2)\\
& x_1 \geq 0 & (3)\\
& x_2 \geq 0 & (4)
\end{aligned}
$$

Current vertex: (3),(4)

Objective value: 0

Move: increase x_1

STOP at $x_1 = 1$

New vertex: (1),(4)

$y_2 = x_2$

$y_1 = 1 - x_1 + x_2$

$$
\begin{aligned}
\text{maximize } & 1 - y_1 + y_2 & \\
\text{subject to } & y_1 \geq 0 & (1)\\
& y_1 \leq 3 & (2)\\
& y_1 - y_2 \leq 1 & (3)\\
& y_2 \geq 0 & (4)
\end{aligned}
$$

Current vertex: (1),(4)

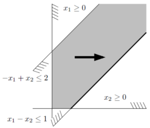
Objective value: 1

Move: increase y_2

STOP at $y_2 = ?$

Another interesting example

Unbounded LP



maximize x_1

subject to

$x_1 - x_2 \leq 1$ (1)

$-x_1 + x_2 \leq 2$ (2)

$x_1 \geq 0$ (3)

$x_2 \geq 0$ (4)

Current vertex: (3),(4)

Objective value: 0

Move: increase x_1

STOP at $x_1 = 1$

New vertex: (1),(4)

$y_2 = x_2$

$y_1 = 1 - x_1 + x_2$

---

maximize $1 - y_1 + y_2$

subject to

$y_1 \geq 0$ (1)

$y_1 \leq 3$ (2)

$y_1 - y_2 \leq 1$ (3)

$y_2 \geq 0$ (4)

Current vertex: (1),(4)

Objective value: 1

Move: increase y_2

STOP at $y_2 = ?$

$\boxed{\text{LP is unbounded}}$

One last example

Degeneracy

$$
\begin{aligned}
\text{maximize } & x_2 & \\
\text{subject to } & -x_1 + x_2 \leq 0 & (1) \\
& x_1 \leq 2 & (2) \\
& x_1 \geq 0 & (3) \\
& x_2 \geq 0 & (4)
\end{aligned}
$$

Current vertex: (3),(4)

One last example

Degeneracy

$$\begin{aligned} \text{maximize } & x_2 \\ \text{subject to } & -x_1 + x_2 \leq 0 && (1) \\ & x_1 \leq 2 && (2) \\ & x_1 \geq 0 && (3) \\ & x_2 \geq 0 && (4) \end{aligned}$$

Current vertex: (3),(4)

Objective value: 0

One last example

Degeneracy

$$
\begin{aligned}
\text{maximize } & x_2 \\
\text{subject to } & -x_1 + x_2 \leq 0 && (1) \\
& x_1 \leq 2 && (2) \\
& x_1 \geq 0 && (3) \\
& x_2 \geq 0 && (4)
\end{aligned}
$$

Current vertex: (3),(4)

Objective value: 0

Move: increase x_2

One last example

Degeneracy

maximize x_2

subject to $-x_1 + x_2 \leq 0$ (1)

$x_1 \leq 2$ (2)

$x_1 \geq 0$ (3)

$x_2 \geq 0$ (4)

Current vertex: (3),(4)

Objective value: 0

Move: increase x_2

One last example

Degeneracy

$$
\begin{aligned}
\text{maximize } & x_2 & \\
\text{subject to } & -x_1 + x_2 \leq 0 & (1)\\
& x_1 \leq 2 & (2)\\
& x_1 \geq 0 & (3)\\
& x_2 \geq 0 & (4)
\end{aligned}
$$

Current vertex: (3),(4)

Objective value: 0

Move: increase x_2

CAN'T!

One last example

Degeneracy

$$\begin{aligned} \text{maximize } & x_2 \\ \text{subject to } & -x_1 + x_2 \leq 0 && (1) \\ & x_1 \leq 2 && (2) \\ & x_1 \geq 0 && (3) \\ & x_2 \geq 0 && (4) \end{aligned}$$

Current vertex: (3),(4)

Objective value: 0

Move: increase x_2

CAN'T!

One last example

Degeneracy

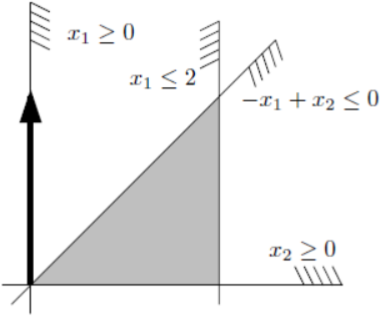
maximize	x_2	
subject to	$-x_1 + x_2 \leq 0$	(1)
	$x_1 \leq 2$	(2)
	$x_1 \geq 0$	(3)
	$x_2 \geq 0$	(4)

Current vertex: (3),(4)

Objective value: 0

Move: increase x_2

CAN'T!



One last example

Degeneracy

maximize x_2

subject to $-x_1 + x_2 \leq 0$ (1)

$x_1 \leq 2$ (2)

$x_1 \geq 0$ (3)

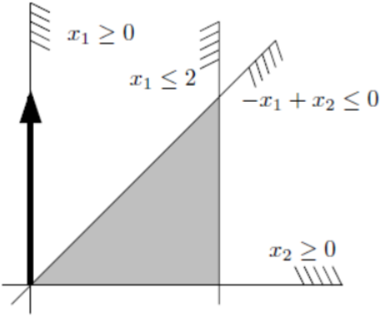
$x_2 \geq 0$ (4)

Current vertex: (3),(4)

Objective value: 0

Move: increase x_2

CAN'T!



Move: increase x_1 $\rightarrow$

LP in standard form

simplex tableau demands . . .

$$\begin{aligned} \text{minimize } & c^\top x \\ \text{subject to } & Ax = b, \\ & x \geq 0. \end{aligned}$$

LP in standard form

simplex tableau demands . . .

$$\begin{aligned} \text{minimize } & c^\top x \\ \text{subject to } & Ax = b, \\ & x \geq 0. \end{aligned}$$

Suppose we are given the inequality constraint

$$x_1 \leq 7.$$

LP in standard form

simplex tableau demands . . .

$$\begin{aligned} \text{minimize } & c^\top x \\ \text{subject to } & Ax = b, \\ & x \geq 0. \end{aligned}$$

Suppose we are given the inequality constraint

$$x_1 \leq 7.$$

Convert to an **equality constraint** – introduce a slack variable:

$$\begin{aligned} x_1 + s_1 &= 7 \\ s_1 &\geq 0. \end{aligned}$$