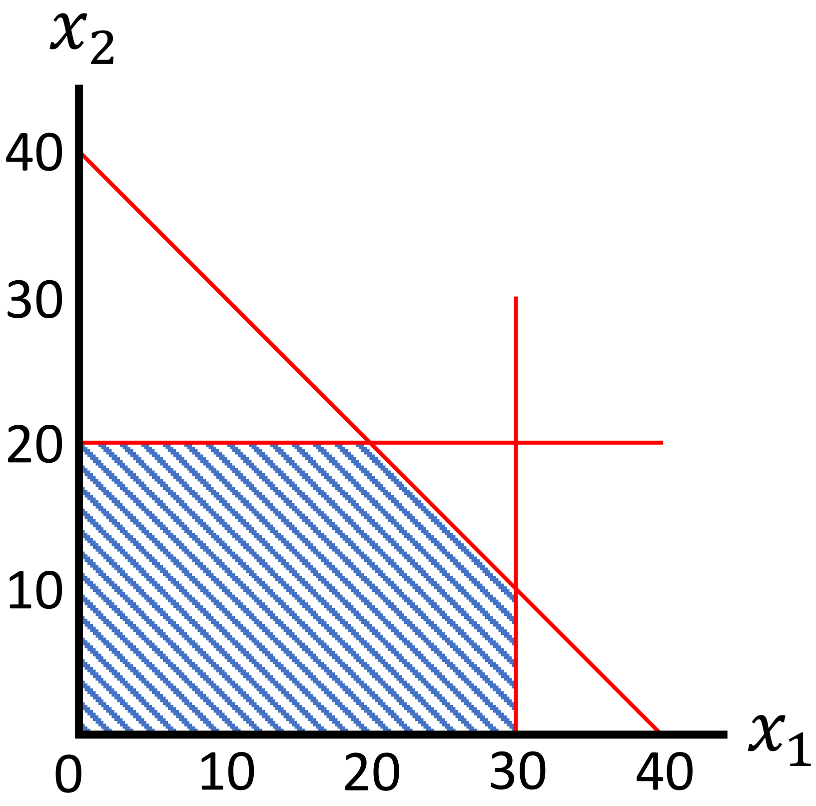
Simplex Algorithm
CSCI 532

Optimal Value



Objective: $\max f(x_1, x_2)$

Subject to: $c_1(x_1, x_2)$

$c_2(x_1, x_2)$

$\vdots$

$c_n(x_1, x_2)$

Properties of optimal solutions:

1. Optimal value occurs at a vertex.
2. Local optimum is global optimum.

Algorithm to find optimal solution:

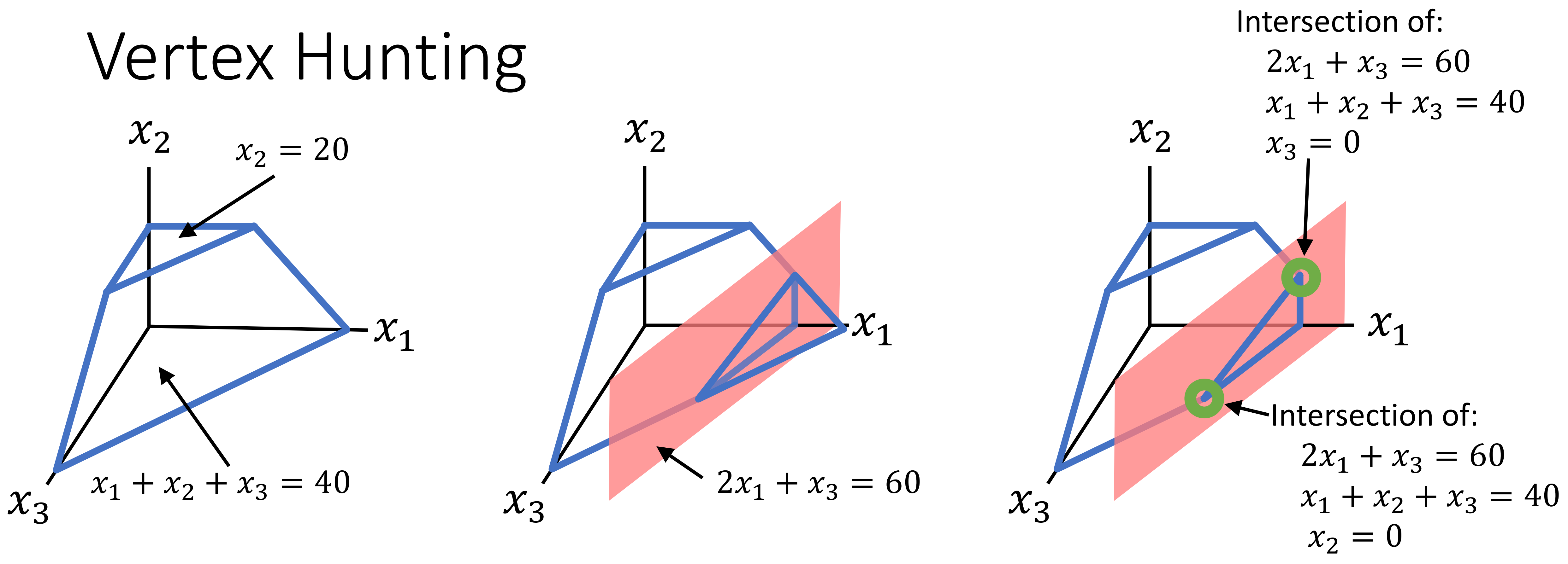
Test each vertex in order until no neighbors have larger (or smaller) value.

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v' with better objective value
      v = v'
   return v
```

How do we find vertices?

Vertex Hunting



Definition: Two vertices are *neighbors* if they share $n - 1$ defining inequalities.

Plan: Move from vertex to vertex by following line formed by intersection of $n - 1$ inequalities.

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v’ with better objective value
      v = v’
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Step 1: Check if current vertex is optimal.

Let $x = (x_1, \dots, x_n)$. Feasible origin ($x = (0, \dots, 0)$) $\Longrightarrow$ vertex, because?

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v’ with better objective value
      v = v’
   return v
```

Objective:	max c^T x
Subject to:	A x $\leq$ b
	x ≥ 0

Step 1: Check if current vertex is optimal.

Let x $= (x_1, \dots, x_n)$. Feasible origin (x $= (0, \dots, 0)$) $\Longrightarrow$ vertex, because it uniquely satisfies n constraints (x ≥ 0).

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v' with better objective value
      v = v'
   return v
```

Objective:	$\max c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Step 1: Check if current vertex is optimal.

How can we tell if the origin is optimal?

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v' with better objective value
      v = v'
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Step 1: Check if current vertex is optimal.

Origin is optimal $\Leftrightarrow c_i \leq 0$, for all i:

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v' with better objective value
      v = v'
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Step 1: Check if current vertex is optimal.

Origin is optimal $\Leftrightarrow c_i \leq 0$, for all i:

If origin is optimal, increasing x_i, for any i will decrease objective

$$\Longrightarrow c_i x_i \geq c_i(x_i + \varepsilon) \Longrightarrow c_i 0 \geq c_i(0 + \varepsilon) \Longrightarrow c_i \leq 0$$

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v' with better objective value
      v = v'
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Step 1: Check if current vertex is optimal.

Origin is optimal $\Leftrightarrow c_i \leq 0$, for all i:

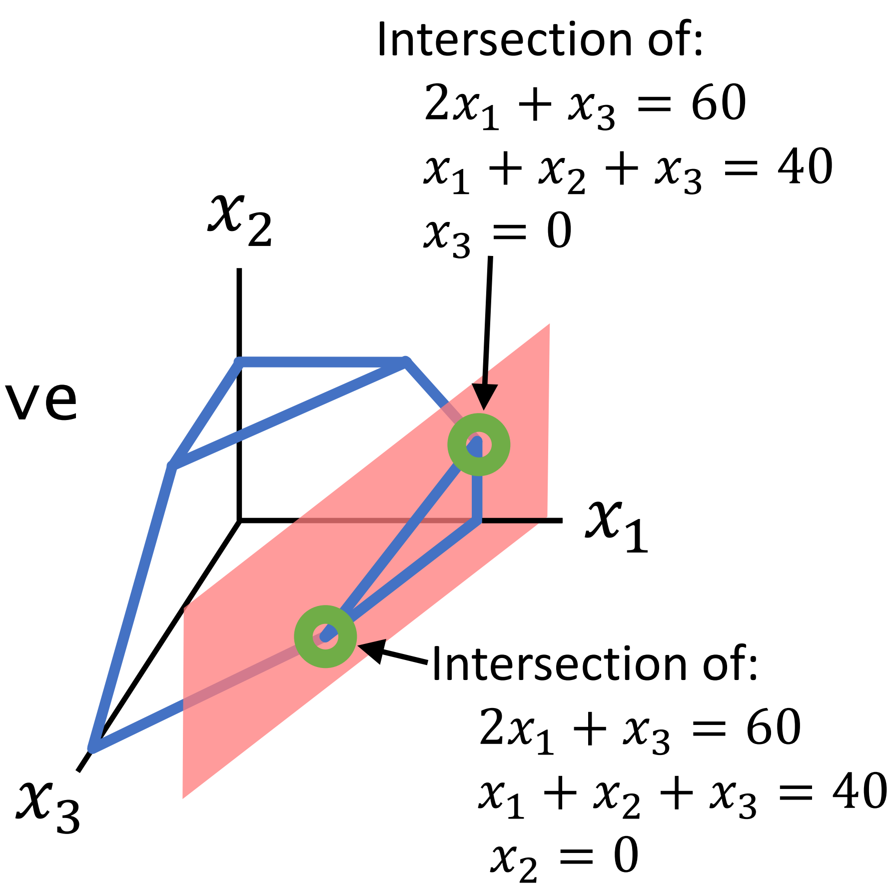
If origin is optimal, increasing x_i, for any i will decrease objective

$$\Longrightarrow c_i x_i \geq c_i(x_i + \varepsilon) \Longrightarrow c_i 0 \geq c_i(0 + \varepsilon) \Longrightarrow c_i \leq 0$$

if $c_i \leq 0$, for all i, $c_i(x_i + \varepsilon) \leq c_i x_i$, i.e., lower objective value

Simplex Algorithm

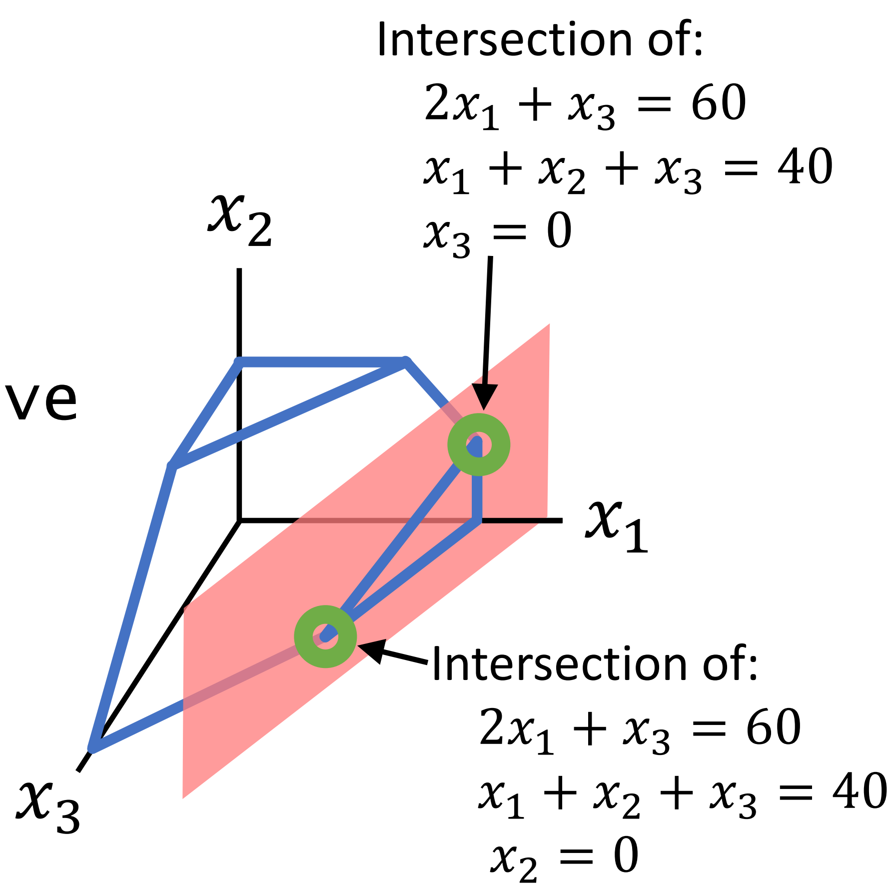
```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v’ with > objective
      v = v’
   return v
```



Step 2: Move to a neighboring vertex.

Simplex Algorithm

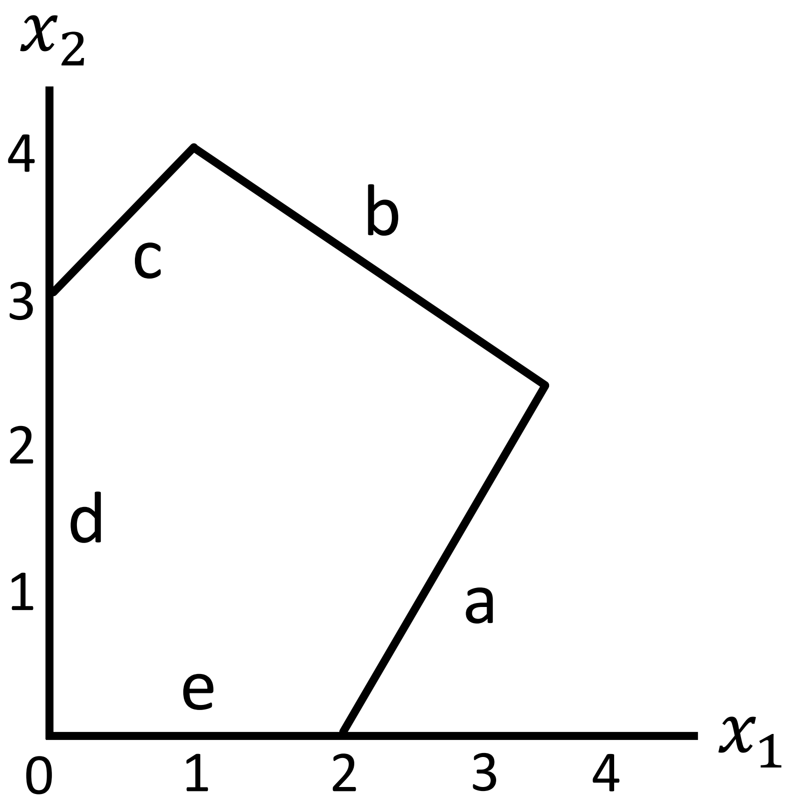
```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



Step 2: Move to a neighboring vertex.

Relax any constraint that allows us to increase the objective function (i.e. x_i, where $c_i > 0$). Keep increasing x_i until another constraint becomes tight.

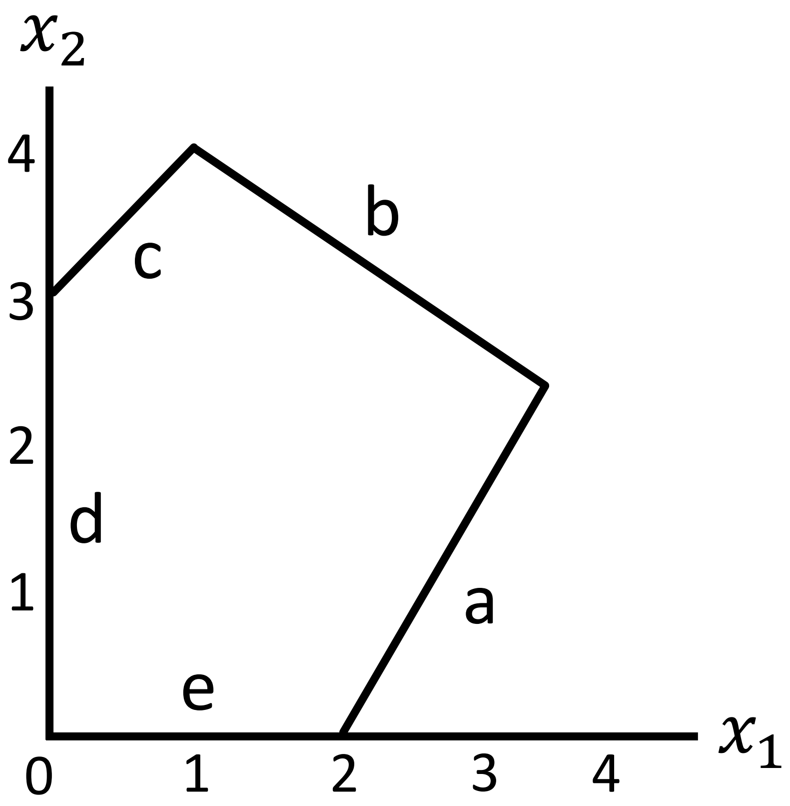
Simplex Algorithm



Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

1. Start at origin, $(x_1, x_2) = (0,0)$.
2. Relax a tight constraint.
3. Stop when another constraint is met.
4. New vertex = intersection of new constraint.

Simplex Algorithm

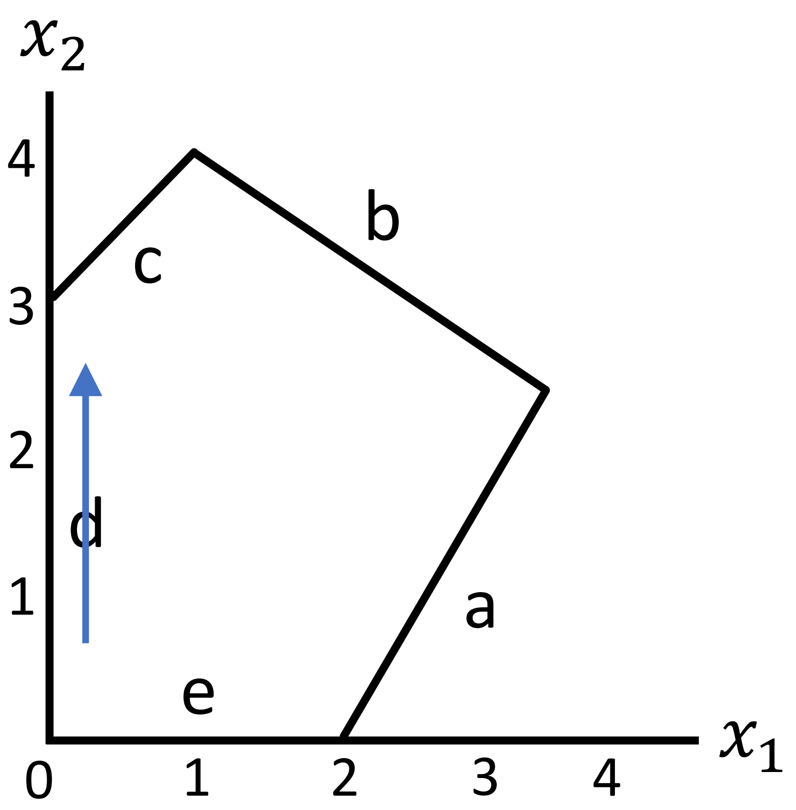


Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

1. Start at origin, $(x_1, x_2) = (0,0)$.
2. Relax a tight constraint.
   Either d or e. Suppose e.
3. Stop when another constraint is met.

4. New vertex = intersection of new constraint.

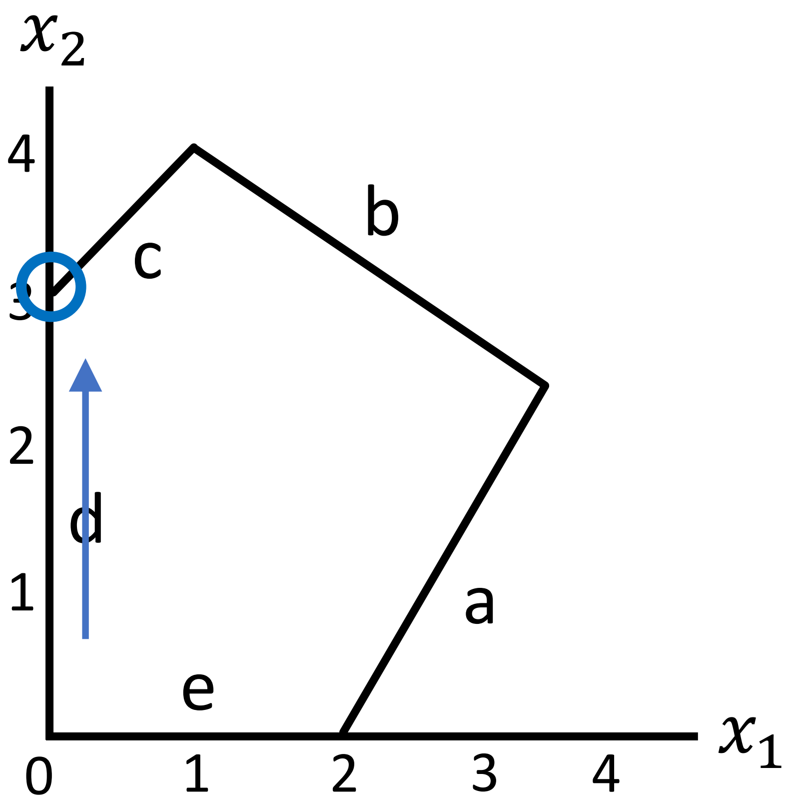
Simplex Algorithm



Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

1. Start at origin, $(x_1, x_2) = (0,0)$.
2. Relax a tight constraint.
   Either d or e. Suppose e.
3. Stop when another constraint is met.
   Let x_2 increase until another constraint is met (a – never, b – 4.5, c – 3)
4. New vertex = intersection of new constraint.

Simplex Algorithm



Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

1. Start at origin, $(x_1, x_2) = (0,0)$.
2. Relax a tight constraint.
   Either d or e. Suppose e.
3. Stop when another constraint is met.
   Let x_2 increase until another constraint is met (a – never, b – 4.5, c – 3)
4. New vertex = intersection of new constraint.
   u = intersection of d and c.

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region of LP
   while ∃ neighbor v' with better objective value
      v = v'
   return v
```

Objective:	$\max c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

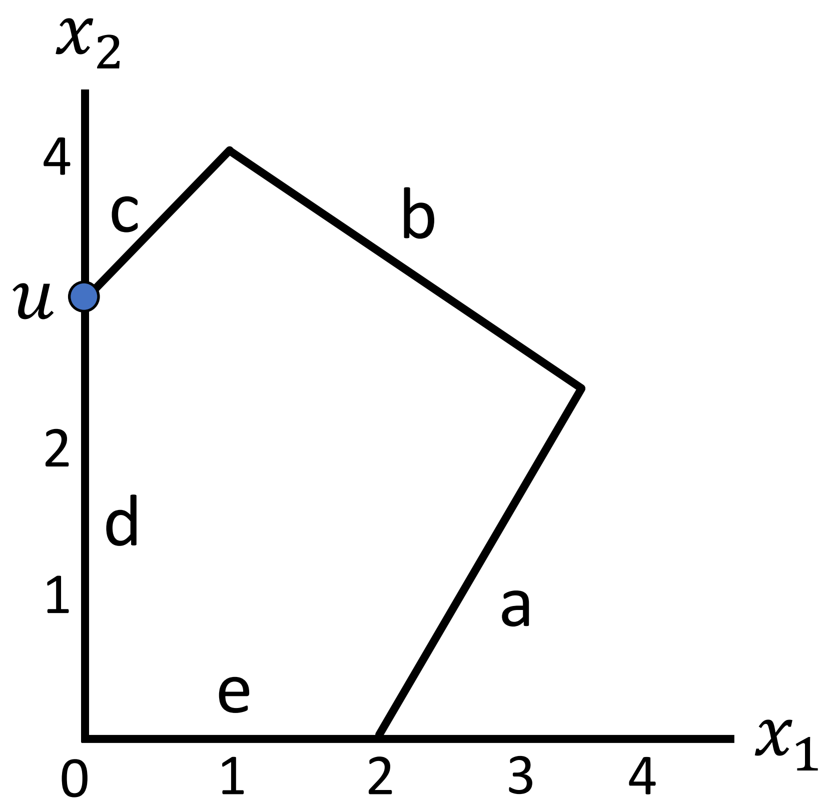
Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

Super easy to do if starting at origin!

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



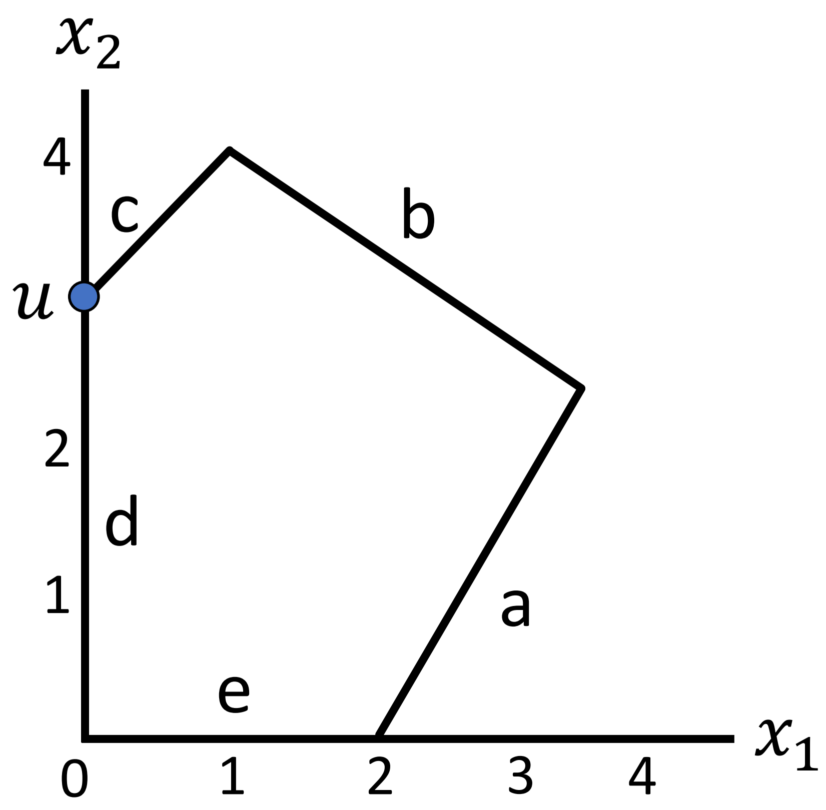
Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

What should we do if we are not at the origin?

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



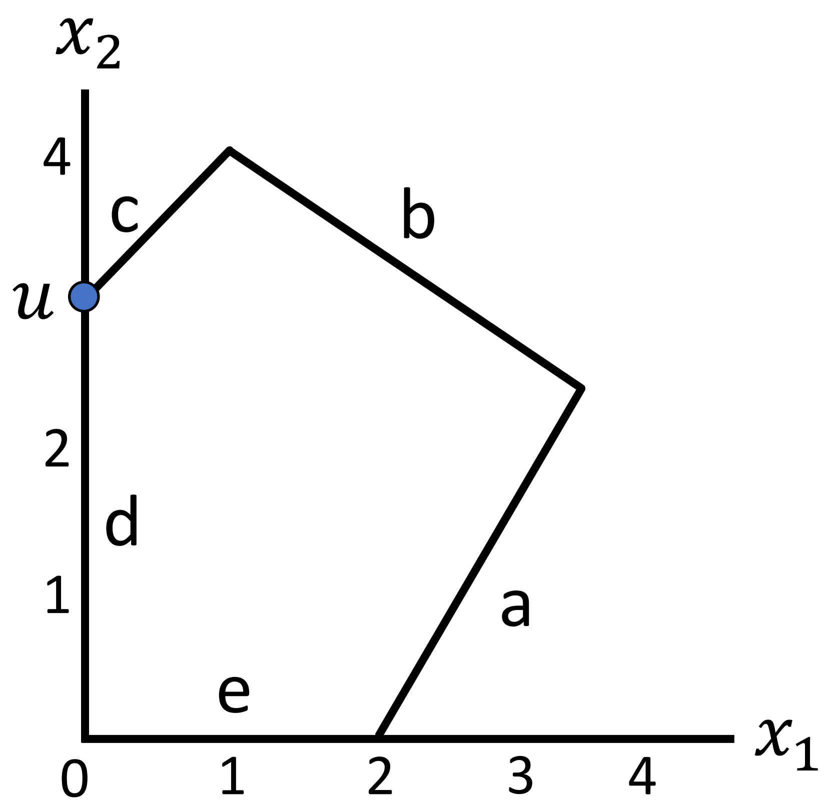
Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

Step 3: Redefine coordinate system so u is the origin.

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

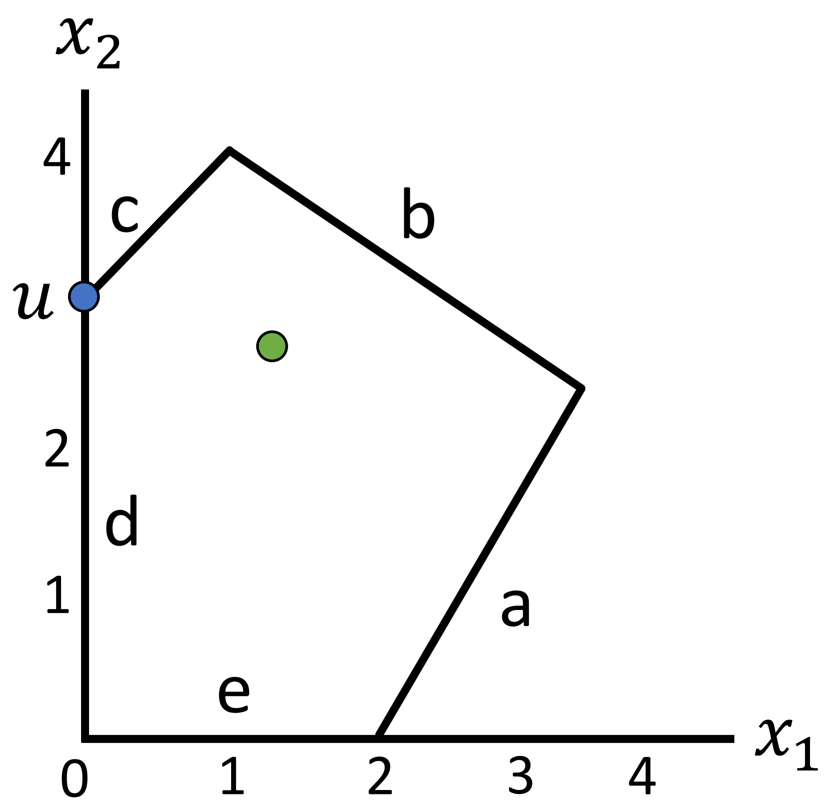
Step 3: Redefine coordinate system so u is the origin.

For constraint $a_i\mathrm{x} \leq b_i$, $y_i = b_i - a_i\mathrm{x}$

distance from x to constraint *i*

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

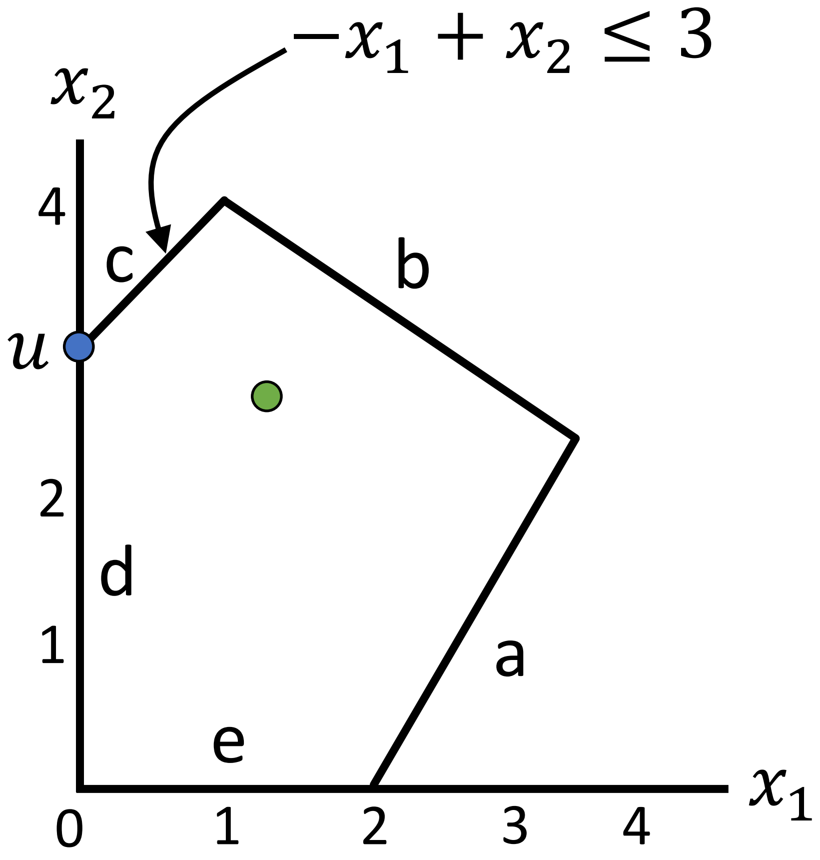
Step 3: Redefine coordinate system so u is the origin.

For constraint $a_i\mathrm{x} \le b_i$, $y_i = b_i - a_i\mathrm{x}$

distance from x to constraint *i*

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

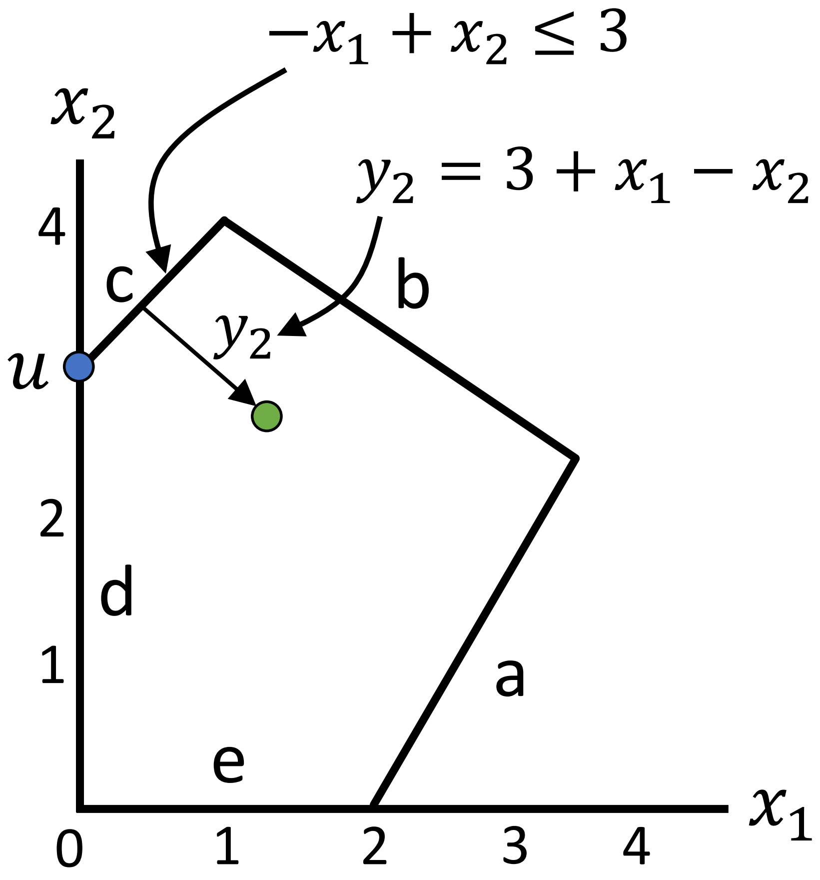
Step 3: Redefine coordinate system so u is the origin.

For constraint $a_i\mathrm{x} \leq b_i$, $y_i = b_i - a_i\mathrm{x}$

distance from x to constraint *i*

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

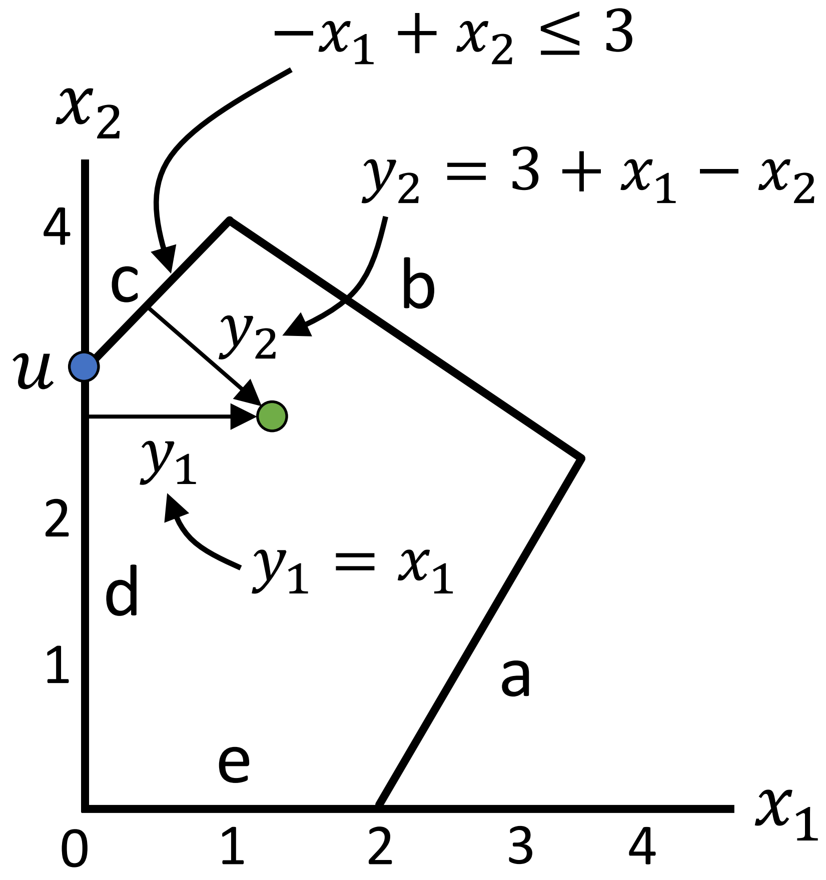
Step 3: Redefine coordinate system so u is the origin.

For constraint $a_i \mathrm{x} \leq b_i$, $y_i = b_i - a_i \mathrm{x}$

distance from x to constraint *i*

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

Step 3: Redefine coordinate system so u is the origin.

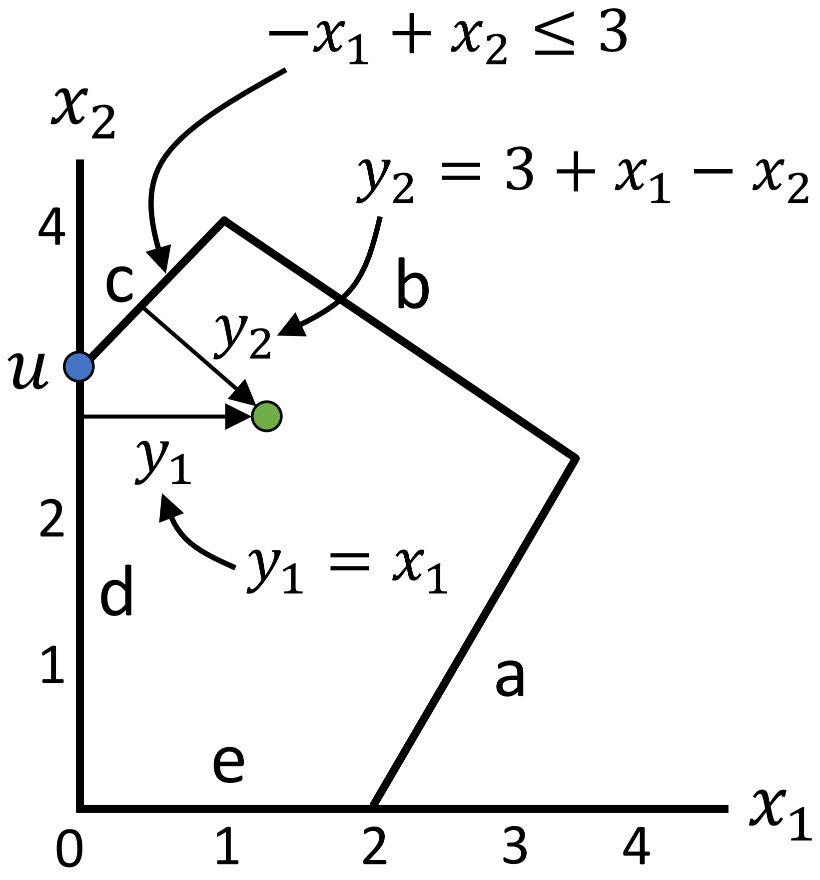
For constraint $a_i\text{x} \leq b_i$, $y_i = b_i - a_i\text{x}$

distance from x to constraint i

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```

$$\boxed{(\boldsymbol{x_1}, \boldsymbol{x_2}) = (\mathbf{0}, \mathbf{3}) \Rightarrow (\boldsymbol{y_1}, \boldsymbol{y_2}) = (\mathbf{0}, \mathbf{0})}$$



Step 1: Check if current vertex is optimal.

Step 2: Move to a neighboring vertex.

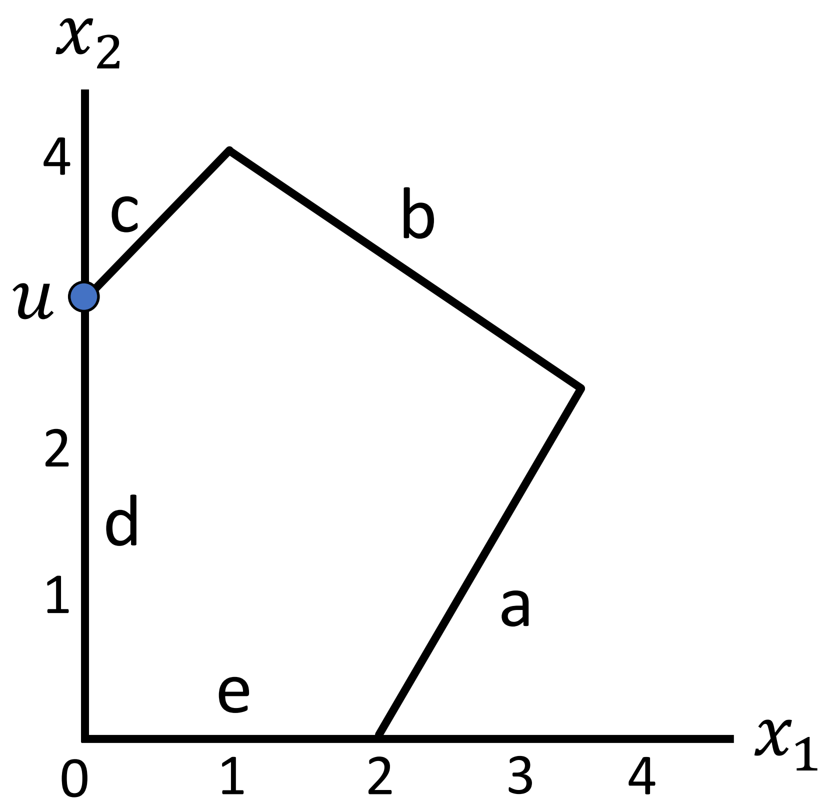
Step 3: Redefine coordinate system so u is the origin.

For constraint $a_i\text{x} \leq b_i$, $y_i = b_i - a_i\text{x}$

distance from x to constraint $\boldsymbol{i}$

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v' with > objective
      v = v'
   return v
```



Step 1: Check if current vertex is optimal.

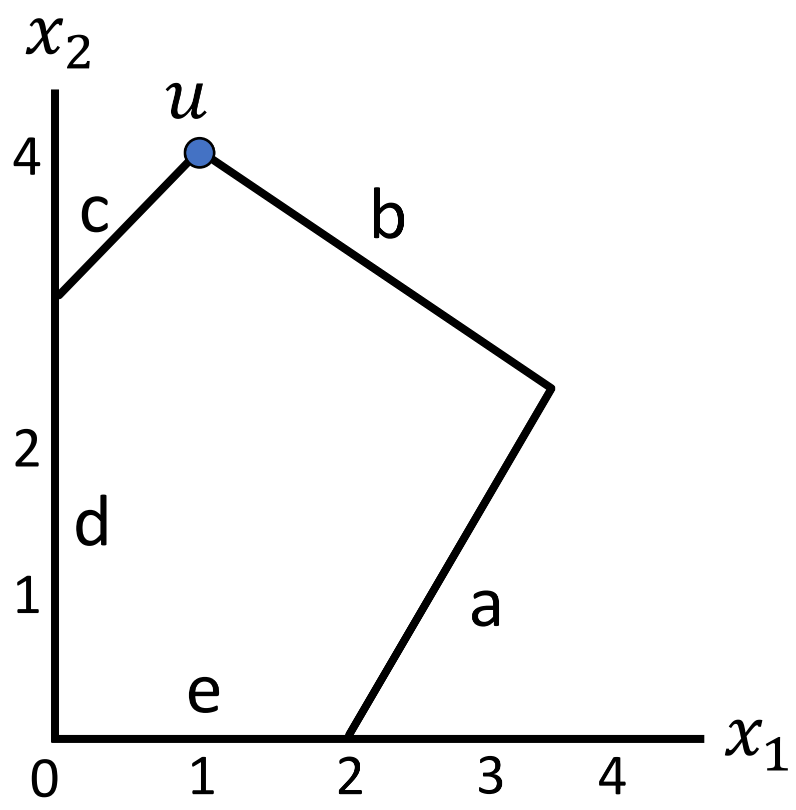
Step 2: Move to a neighboring vertex.

Step 3: Redefine coordinate system so u is the origin.

Step 4: ???

Simplex Algorithm

```
Simplex(LP)
   v = vertex in feasible region
   while ∃ neighbor v’ with > objective
      v = v’
   return v
```



Step 1: Check if current vertex is optimal.

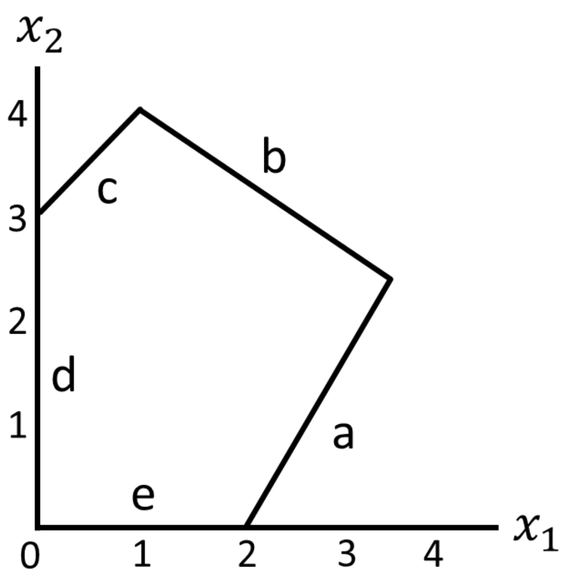
Step 2: Move to a neighboring vertex.

Step 3: Redefine coordinate system so u is the origin.

Step 4: Repeat.

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

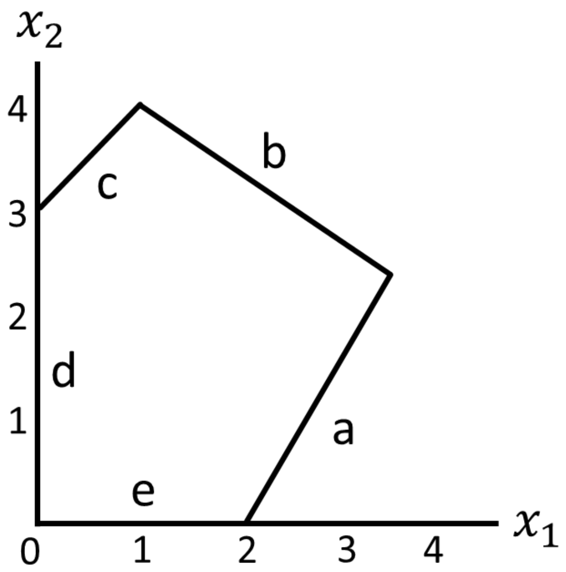
Vertex:

Objective Value:

Relax:

Stop at:

Vertex Local Coordinates:

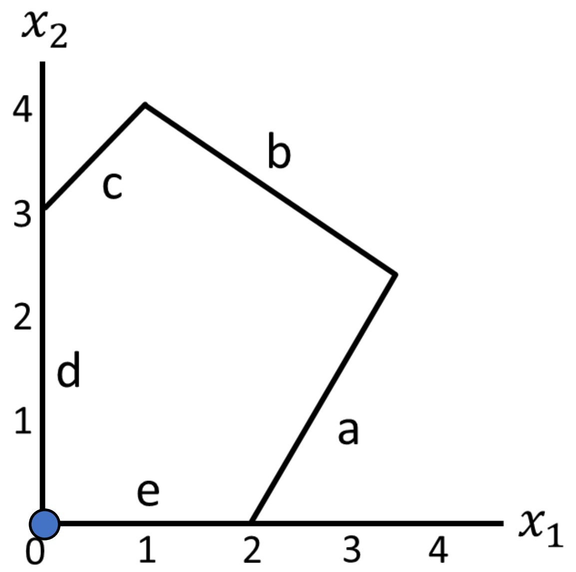


Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: Origin (d,e)
Objective Value:
Relax:
Stop at:
Vertex Local Coordinates:

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: Origin (d,e)

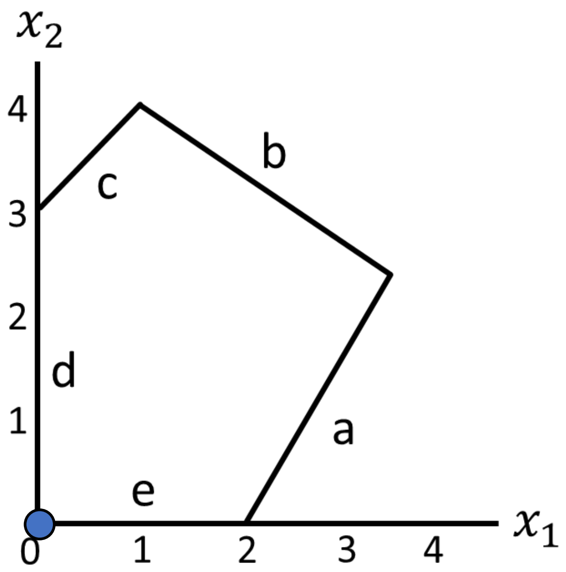
Objective Value: 0

Relax:

Stop at:

Vertex Local Coordinates:

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: Origin (d,e)

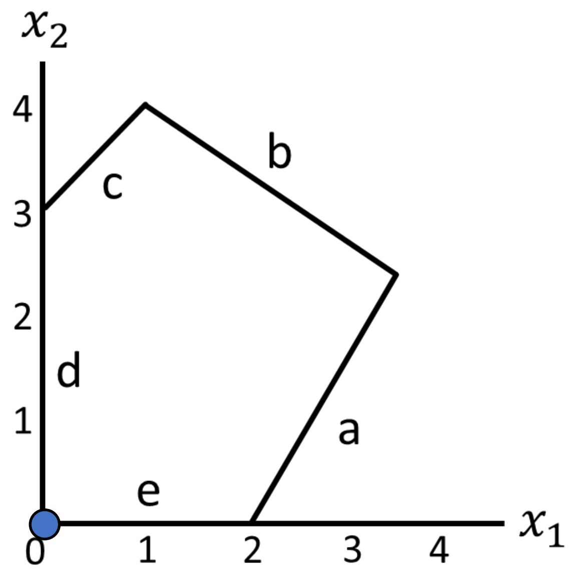
Objective Value: 0

Relax: e

Stop at:

Vertex Local Coordinates:

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: Origin (d,e)

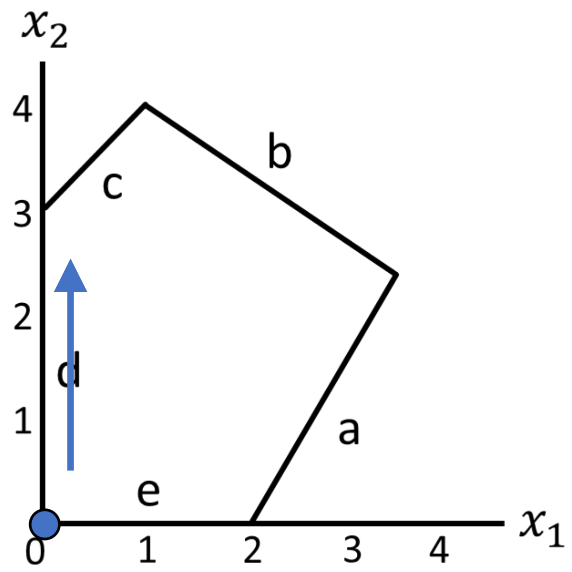
Objective Value: 0

Relax: e

Stop at:

Vertex Local Coordinates:

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: Origin (d,e)

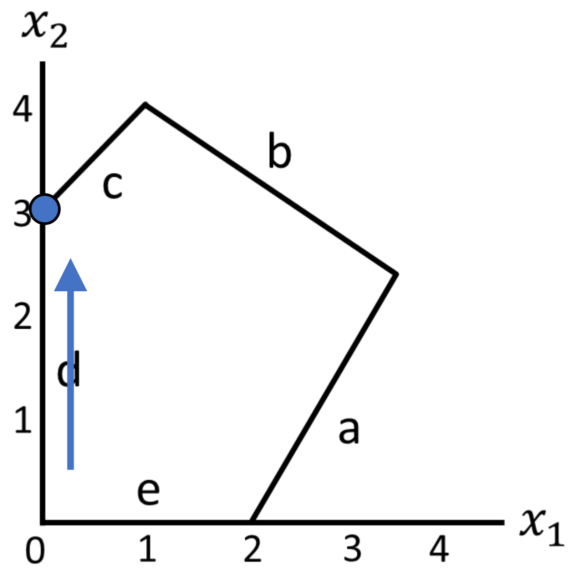
Objective Value: 0

Relax: e

Stop at: c,d

Vertex Local Coordinates:

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

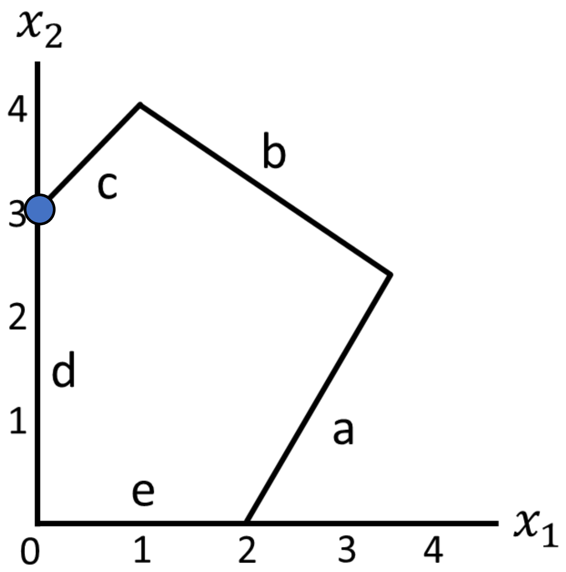
Objective Value: 0

Relax: e

Stop at: c,d

Vertex Local Coordinates:

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: 0

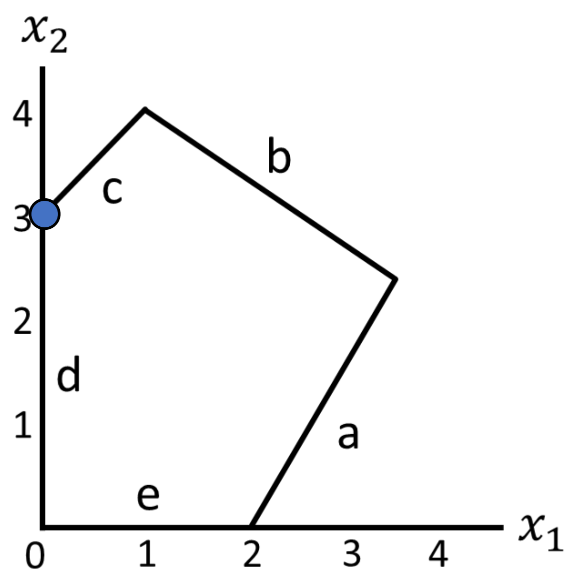
Relax: e

Stop at: c,d

Vertex Local Coordinates:

$$x_1 \geq 0 \Longrightarrow -x_1 \leq 0 \Longrightarrow y_1 = x_1$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: 0

Relax: e

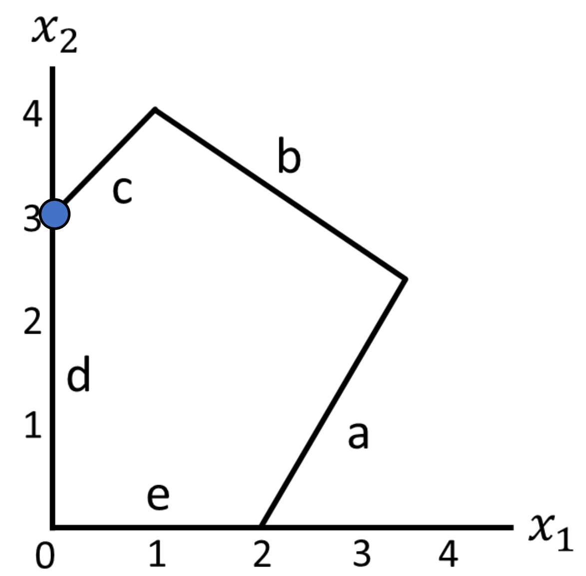
Stop at: c,d

Vertex Local Coordinates:

$$x_1 \geq 0 \Longrightarrow -x_1 \leq 0 \Longrightarrow y_1 = x_1$$

$$-x_1 + x_2 \leq 3 \Longrightarrow y_2 = 3 + x_1 - x_2$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi - ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: 0

Relax: e

Stop at: c,d

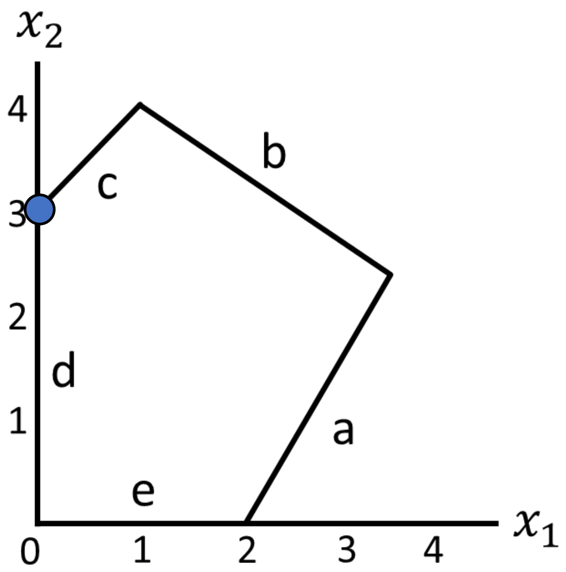
Vertex Local Coordinates:

$$x_1 \geq 0 \Longrightarrow -x_1 \leq 0 \Longrightarrow y_1 = x_1$$

$$-x_1 + x_2 \leq 3 \Longrightarrow y_2 = 3 + x_1 - x_2$$

$$\Longrightarrow x_1 = y_1, x_2 = 3 + y_1 - y_2$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi - ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: 0

Relax: e

Stop at: c,d

Vertex Local Coordinates:

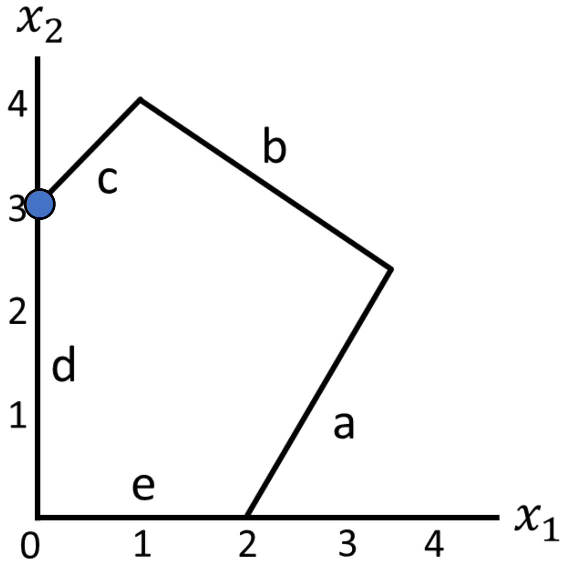
$$x_1 \geq 0 \Longrightarrow -x_1 \leq 0 \Longrightarrow y_1 = x_1$$

$$-x_1 + x_2 \leq 3 \Longrightarrow y_2 = 3 + x_1 - x_2$$

$$\Longrightarrow x_1 = y_1, x_2 = 3 + y_1 - y_2$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

Objective:	max	
Subject to:		(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: 0

Relax: e

Stop at: c,d

Vertex Local Coordinates:

$$x_1 \geq 0 \Rightarrow -x_1 \leq 0 \Rightarrow y_1 = x_1$$

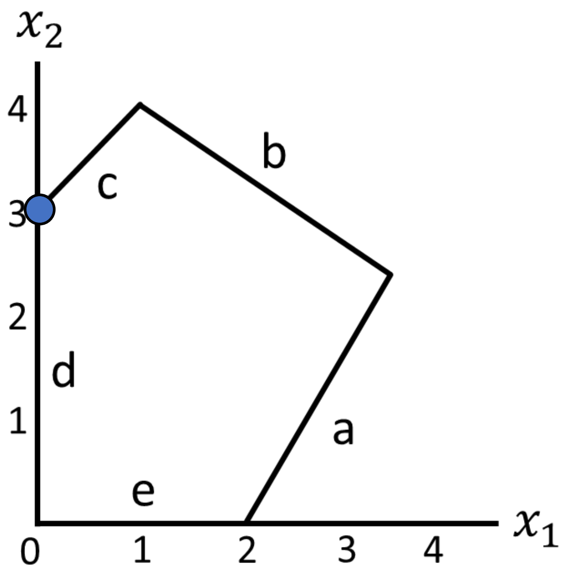
$$-x_1 + x_2 \leq 3 \Rightarrow y_2 = 3 + x_1 - x_2$$

$$\Rightarrow x_1 = y_1, x_2 = 3 + y_1 - y_2$$

$$2x_1 + 5x_2 = 2y_1 + 15 + 5y_1 - 5y_2 = 15 + 7y_1 - 5y_2$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:		(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: ~~0~~ 15

Relax: e

Stop at: c,d

Vertex Local Coordinates:

$$x_1 \geq 0 \Rightarrow -x_1 \leq 0 \Rightarrow y_1 = x_1$$

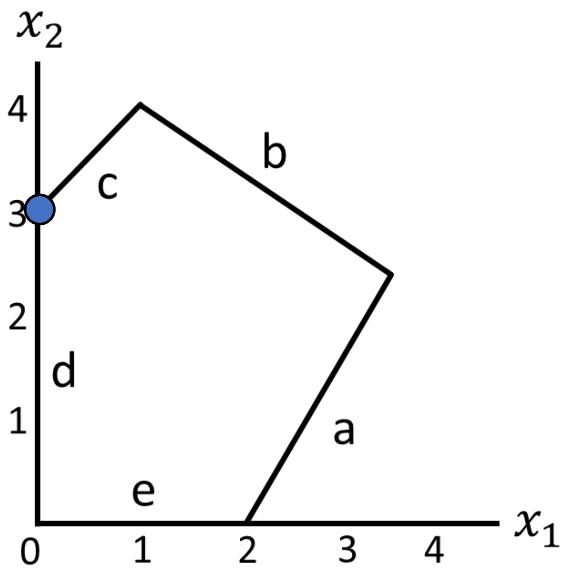
$$-x_1 + x_2 \leq 3 \Rightarrow y_2 = 3 + x_1 - x_2$$

$$\Rightarrow x_1 = y_1, x_2 = 3 + y_1 - y_2$$

$$2x_1 + 5x_2 = 2y_1 + 15 + 5y_1 - 5y_2 = 15 + 7y_1 - 5y_2$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:		(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi - ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: ~~0~~ 15

Relax: e

Stop at: c,d

Vertex Local Coordinates:

$$x_1 \geq 0 \Rightarrow -x_1 \leq 0 \Rightarrow y_1 = x_1$$

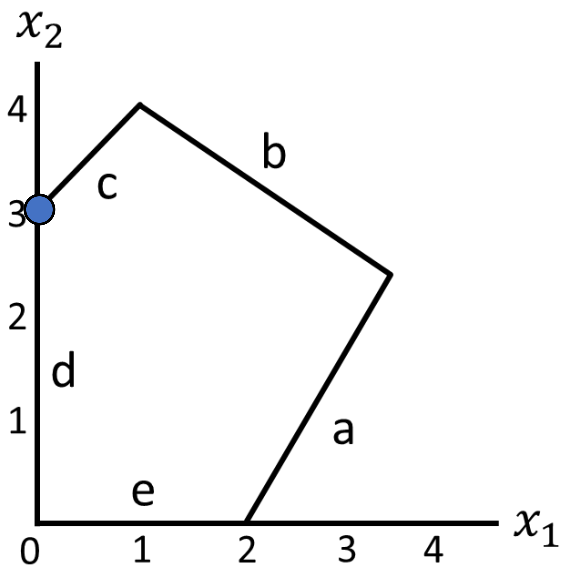
$$-x_1 + x_2 \leq 3 \Rightarrow y_2 = 3 + x_1 - x_2$$

$$\Rightarrow x_1 = y_1, x_2 = 3 + y_1 - y_2$$

$$2x_1 - x_2 = 2y_1 - 3 - y_1 + y_2 = y_1 + y_2 - 3$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi - ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: ~~Origin (d,e)~~ c,d

Objective Value: ~~0~~ 15

Relax: e

Stop at: c,d

Vertex Local Coordinates:

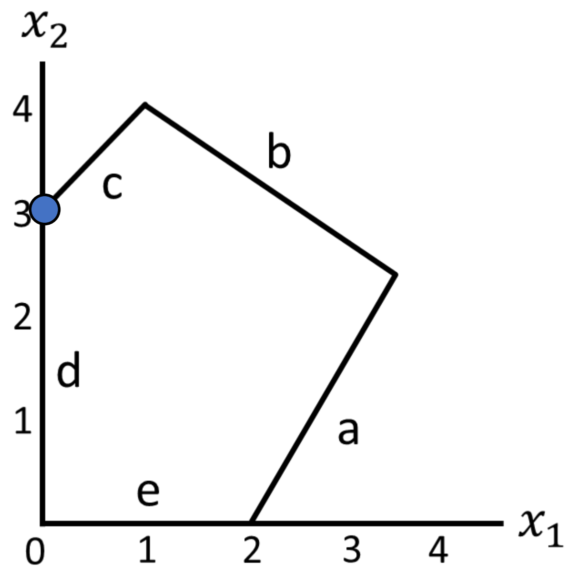
$$x_1 \geq 0 \Rightarrow -x_1 \leq 0 \Rightarrow y_1 = x_1$$

$$-x_1 + x_2 \leq 3 \Rightarrow y_2 = 3 + x_1 - x_2$$

$$\Rightarrow x_1 = y_1, x_2 = 3 + y_1 - y_2$$

Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: c,d

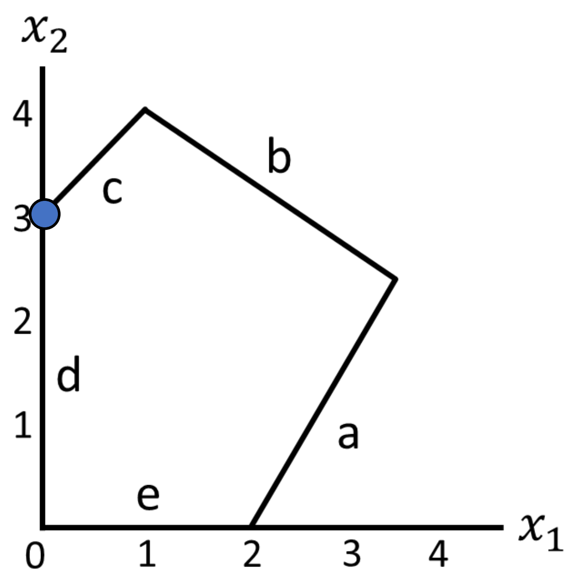
Objective Value: 15

Relax:

Stop at:

Vertex Local Coordinates:

Objective:	$\max 15 + 7y_1 - 5y_2$
Subject to:	$y_1 + y_2 \leq 7$ (a)
	$3y_1 - 2y_2 \leq 3$ (b)
	$y_2 \geq 0$ (c)
	$y_1 \geq 0$ (d)
	$-y_1 + y_2 \leq 3$ (e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: c,d

Objective Value: 15

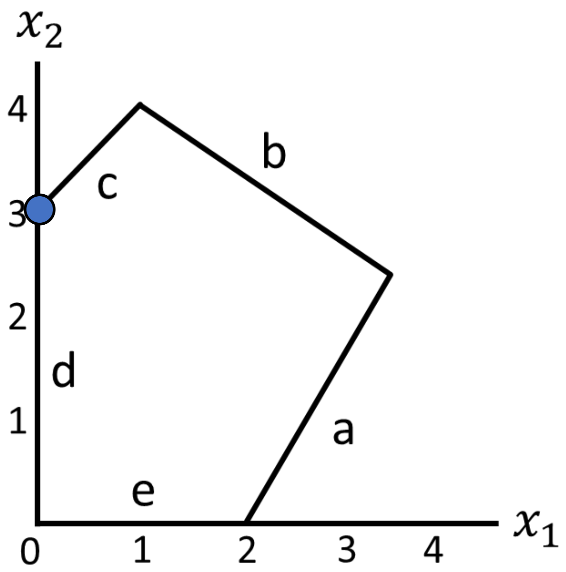
Relax:

Stop at:

Vertex Local Coordinates:

Which constraint should we relax?

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: c,d

Objective Value: 15

Relax: d

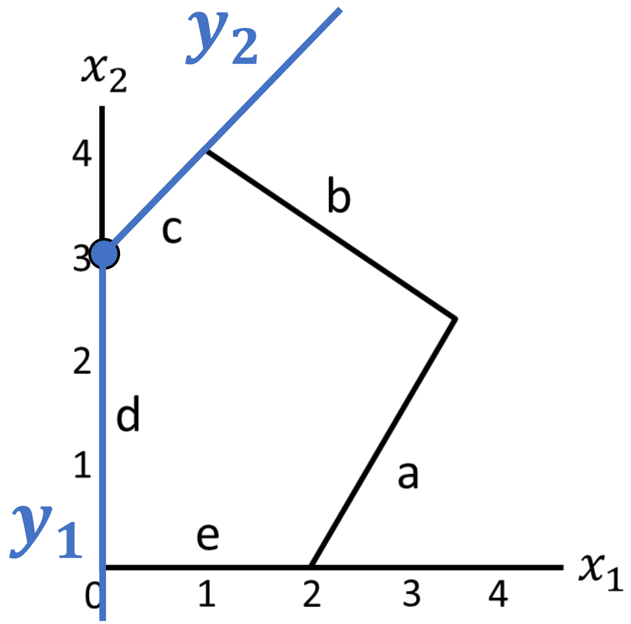
Stop at:

Vertex Local Coordinates:

Which constraint should we relax?

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)

Increasing y_2 worsens objective. Therefore, we should increase y_1, which is constraint (d).



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Vertex: c,d

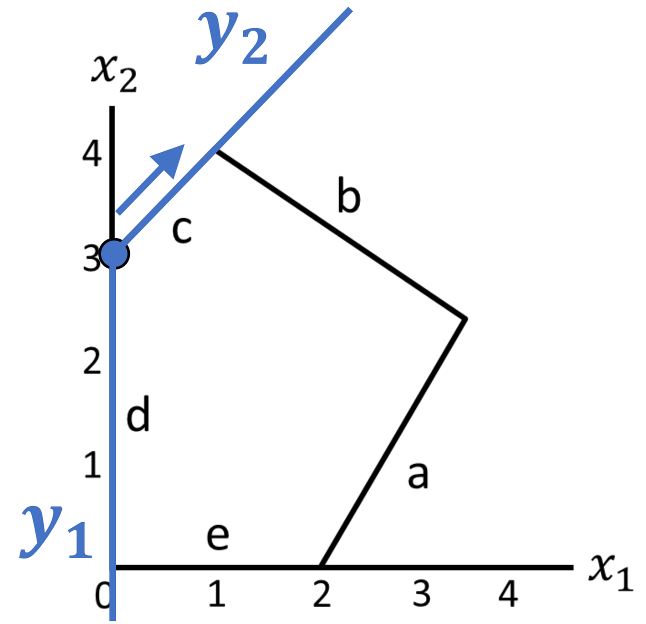
Objective Value: 15

Relax: d

Stop at:

Vertex Local Coordinates:

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: c,d

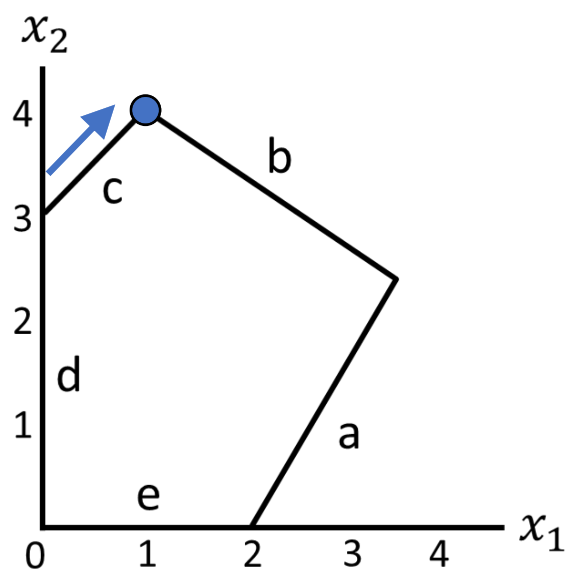
Objective Value: 15

Relax: d

Stop at: b,c

Vertex Local Coordinates:

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Vertex: ~~c,d~~ b,c

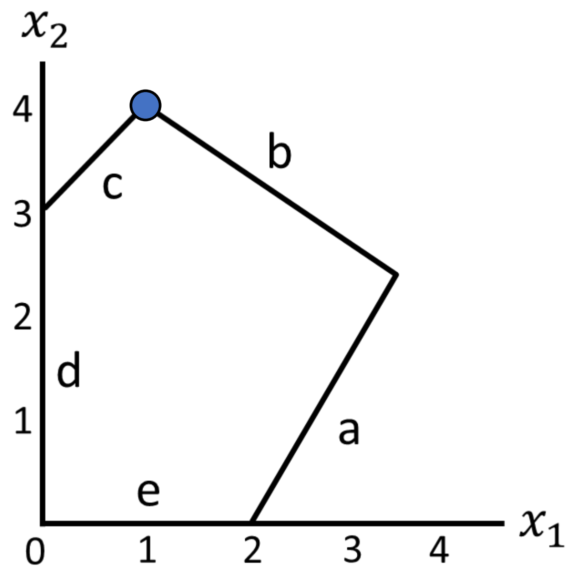
Objective Value: 15

Relax: d

Stop at: b,c

Vertex Local Coordinates:

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·y ≤ b_i
         z_i = b_i – a_i·y
      reformulate LP in terms of z_i
   return v
```

Vertex: ~~c,d~~ b,c

Objective Value: 15

Relax: d

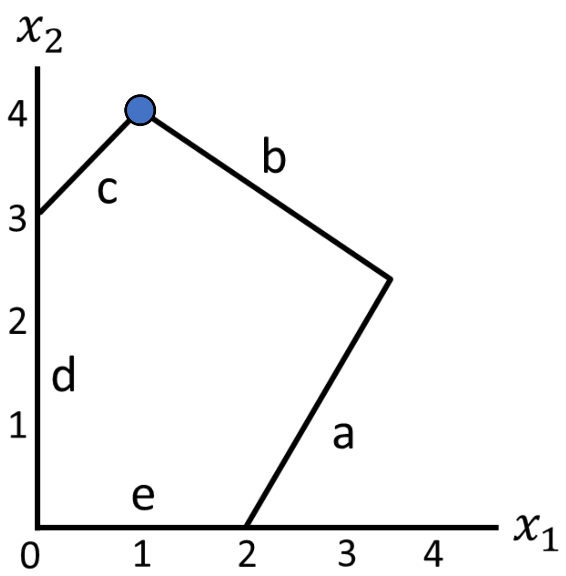
Stop at: b,c

Vertex Local Coordinates:

$$3y_1 - 2y_2 \leq 3 \Longrightarrow z_1 = 3 - 3y_1 + 2y_2$$

$$y_2 \geq 0 \Longrightarrow z_2 = y_2$$

$$\Longrightarrow y_1 = 1 - \frac{1}{3}z_1 + \frac{2}{3}z_2, y_2 = z_2$$



Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·y ≤ b_i
         z_i = b_i - a_i·y
      reformulate LP in terms of z_i
   return v
```

Vertex: ~~c,d~~ b,c

Objective Value: 15

Relax: d

Stop at: b,c

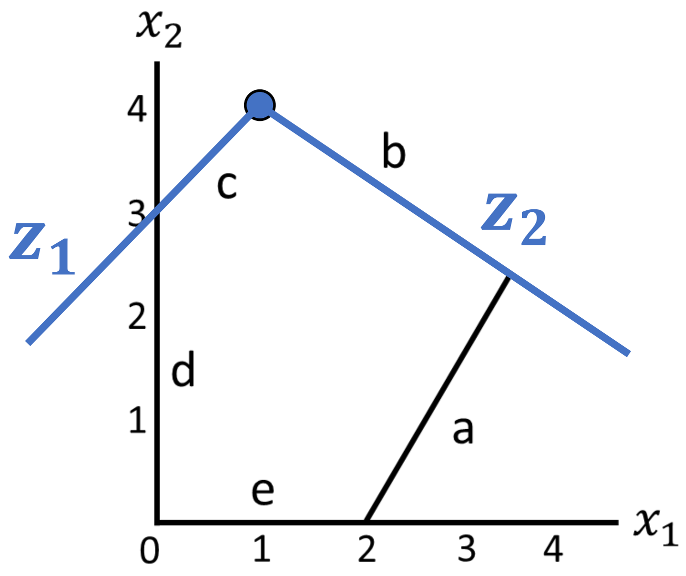
Vertex Local Coordinates:

$$3y_1 - 2y_2 \leq 3 \Longrightarrow z_1 = 3 - 3y_1 + 2y_2$$

$$y_2 \geq 0 \Longrightarrow z_2 = y_2$$

$$\Longrightarrow y_1 = 1 - \frac{1}{3}z_1 + \frac{2}{3}z_2, y_2 = z_2$$

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·y ≤ bi
         zi = bi - ai·y
      reformulate LP in terms of zi
   return v
```

Vertex: ~~c,d~~ b,c

Objective Value: 15

Relax: d

Stop at: b,c

Vertex Local Coordinates:

$3y_1 - 2y_2 \leq 3 \Longrightarrow z_1 = 3 - 3y_1 + 2y_2$

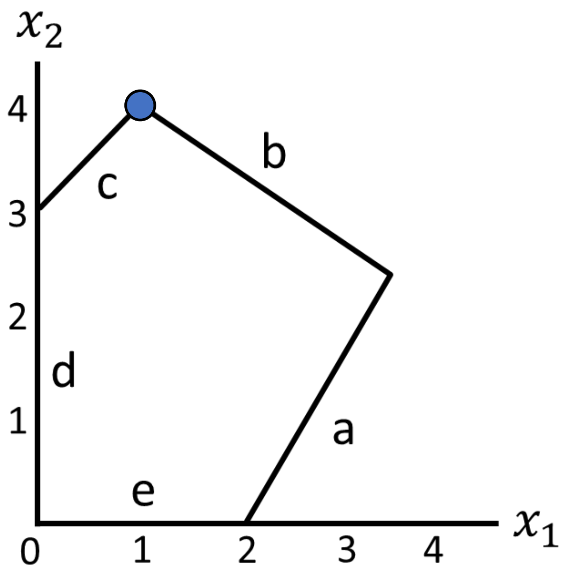
$y_2 \geq 0 \Longrightarrow z_2 = y_2$

$\Longrightarrow y_1 = 1 - \frac{1}{3}z_1 + \frac{2}{3}z_2, y_2 = z_2$

$$15 + 7y_1 - 5y_2 = 15 + 7 - \frac{7}{3}z_1 + \frac{14}{3}z_2 - 5z_2 = 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$$

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)

Objective:	$\max 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$	
Subject to:		(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·y ≤ b_i
         z_i = b_i - a_i·y
      reformulate LP in terms of z_i
   return v
```

Vertex: ~~c,d~~ b,c

Objective Value: ~~15~~ 22

Relax: d

Stop at: b,c

Vertex Local Coordinates:

$$3y_1 - 2y_2 \le 3 \Longrightarrow z_1 = 3 - 3y_1 + 2y_2$$

$$y_2 \ge 0 \Longrightarrow z_2 = y_2$$

$$\Longrightarrow y_1 = 1 - \tfrac{1}{3}z_1 + \tfrac{2}{3}z_2, y_2 = z_2$$

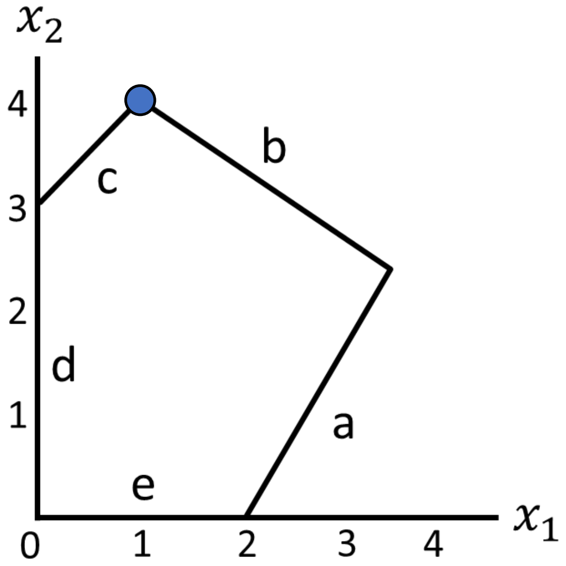
$$15 + 7y_1 - 5y_2 = 15 + 7 - \frac{7}{3}z_1 + \frac{14}{3}z_2 - 5z_2 = 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$$

Objective: $\max 15 + 7y_1 - 5y_2$

Subject to:
- $y_1 + y_2 \le 7$ (a)
- $3y_1 - 2y_2 \le 3$ (b)
- $y_2 \ge 0$ (c)
- $y_1 \ge 0$ (d)
- $-y_1 + y_2 \le 3$ (e)

Objective: $\max 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$

Subject to:
- (a)
- (b)
- (c)
- (d)
- (e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·y ≤ b_i
         z_i = b_i - a_i·y
      reformulate LP in terms of z_i
   return v
```

Vertex: ~~c,d~~ b,c

Objective Value: ~~15~~ 22

Relax: d

Stop at: b,c

Vertex Local Coordinates:

$$3y_1 - 2y_2 \leq 3 \Longrightarrow z_1 = 3 - 3y_1 + 2y_2$$

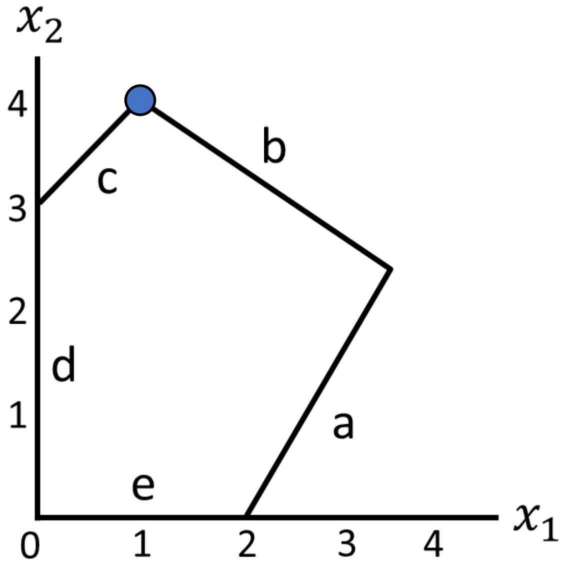
$$y_2 \geq 0 \Longrightarrow z_2 = y_2$$

$$\Longrightarrow y_1 = 1 - \frac{1}{3}z_1 + \frac{2}{3}z_2, y_2 = z_2$$

$$15 + 7y_1 - 5y_2 = 15 + 7 - \frac{7}{3}z_1 + \frac{14}{3}z_2 - 5z_2 = 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$$

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)

Objective:	$\max 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$	
Subject to:		(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·y ≤ b_i
         z_i = b_i - a_i·y
      reformulate LP in terms of z_i
   return v
```

Vertex: ~~c,d~~ b,c

Objective Value: ~~15~~ 22

Relax: d

Stop at: b,c

Vertex Local Coordinates:

$$3y_1 - 2y_2 \le 3 \Longrightarrow z_1 = 3 - 3y_1 + 2y_2$$

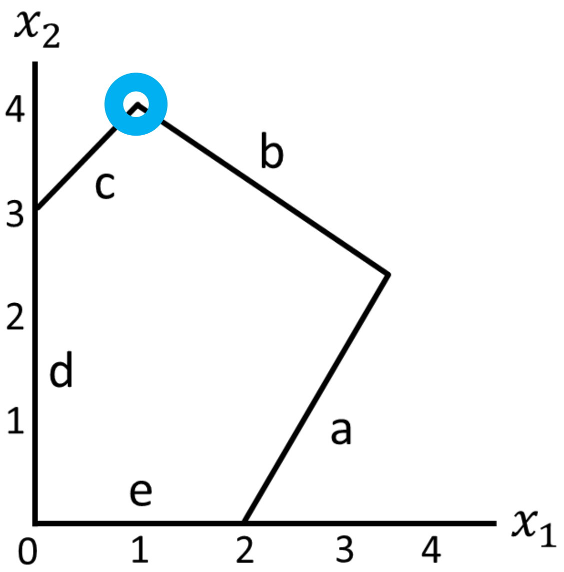
$$y_2 \ge 0 \Longrightarrow z_2 = y_2$$

$$\Longrightarrow y_1 = 1 - \frac{1}{3}z_1 + \frac{2}{3}z_2, y_2 = z_2$$

$$15 + 7y_1 - 5y_2 = 15 + 7 - \frac{7}{3}z_1 + \frac{14}{3}z_2 - 5z_2 = 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$$

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \le 7$	(a)
	$3y_1 - 2y_2 \le 3$	(b)
	$y_2 \ge 0$	(c)
	$y_1 \ge 0$	(d)
	$-y_1 + y_2 \le 3$	(e)

Objective:	$\max 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$	
Subject to:		(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·y ≤ b_i
         z_i = b_i - a_i·y
      reformulate LP in terms of z_i
   return v
```

Vertex: ~~c,d~~ b,c

Objective Value: ~~15~~ 22

Relax: d

Stop at: b,c

Vertex Local Coordinates:

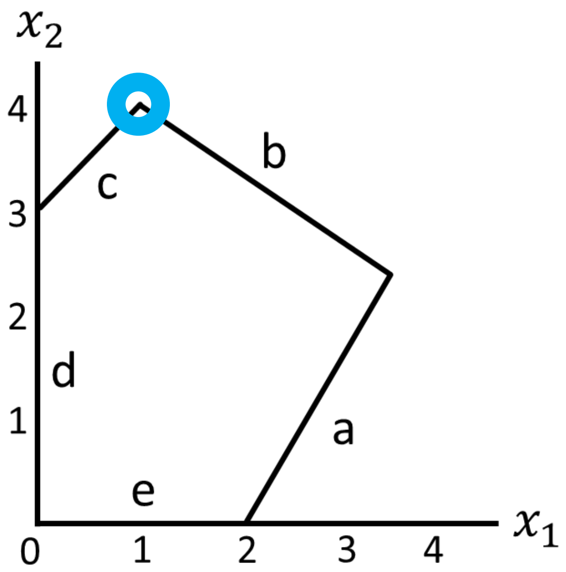
$$3y_1 - 2y_2 \le 3 \Longrightarrow z_1 = 3 - 3y_1 + 2y_2$$

$$y_2 \ge 0 \Longrightarrow z_2 = y_2$$

$$\Longrightarrow y_1 = 1 - \frac{1}{3}z_1 + \frac{2}{3}z_2, y_2 = z_2$$

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \le 7$	(a)
	$3y_1 - 2y_2 \le 3$	(b)
	$y_2 \ge 0$	(c)
	$y_1 \ge 0$	(d)
	$-y_1 + y_2 \le 3$	(e)

Objective:	$\max 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$	
Subject to:		(a)
		(b)
		(c)
		(d)
		(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·y ≤ bi
         zi = bi – ai·y
      reformulate LP in terms of zi
   return v
```

Loose Ends:

1. Starting Vertex
2. Unbounded/infeasible solution
3. Degenerate vertices
4. Running Time

Objective: $\max 15 + 7y_1 - 5y_2$

Subject to:

$y_1 + y_2 \leq 7$ (a)

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

$y_2 \geq 0$ (c)

$y_1 \geq 0$ (d)

$-y_1 + y_2 \leq 3$ (e)

Objective: $\max 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$

Subject to:

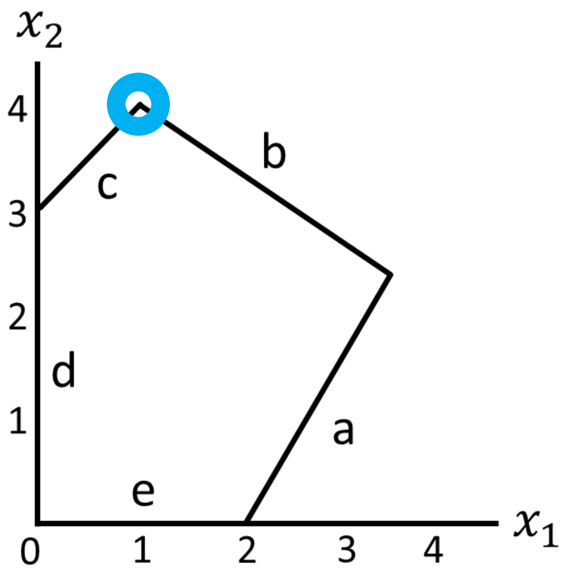
(a)

(b)

(c)

(d)

(e)



```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Objective:	max $c^T x$
Subject to:	$A\,x \leq b$
	$x \geq 0$

Running Time ($n =$ # variables, $m =$ # constraints defined by A):

```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Objective: $\max c^T x$
Subject to: $A x \leq b$
$x \geq 0$

Running Time ($n =$ # variables, $m =$ # constraints defined by A):

- Number of possible neighbors for a given vertex?

```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi - ai·x
      reformulate LP in terms of yi
   return v
```

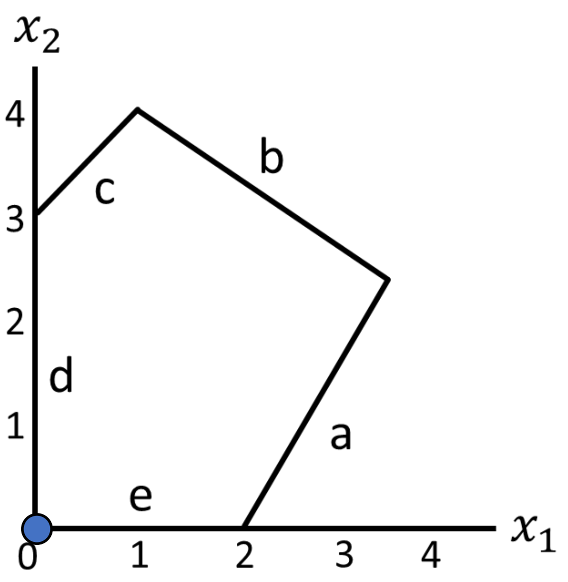
Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

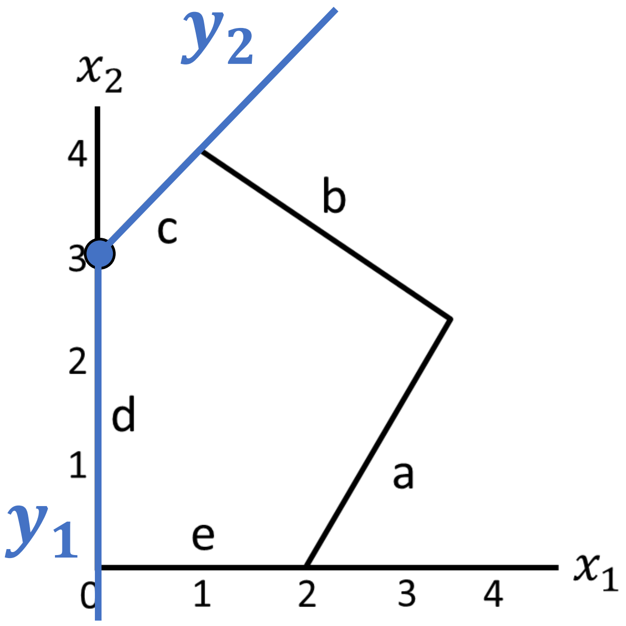
Running Time ($n =$ # variables, $m =$ # constraints defined by A):

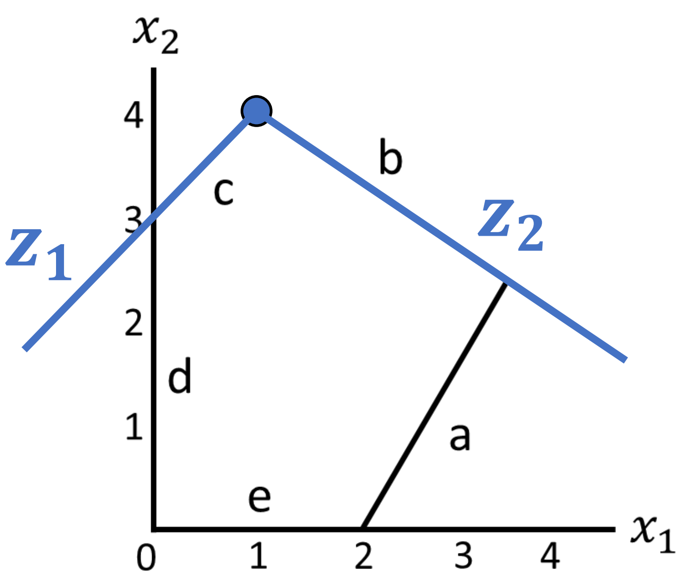
- Number of possible neighbors for a given vertex?

n non-negativity constraints are forming the vertex

n non-negativity constraints are forming the vertex







Objective:	$\max 2x_1 + 5x_2$	
Subject to:	$2x_1 - x_2 \leq 4$	(a)
	$x_1 + 2x_2 \leq 9$	(b)
	$-x_1 + x_2 \leq 3$	(c)
	$x_1 \geq 0$	(d)
	$x_2 \geq 0$	(e)

Objective:	$\max 15 + 7y_1 - 5y_2$	
Subject to:	$y_1 + y_2 \leq 7$	(a)
	$3y_1 - 2y_2 \leq 3$	(b)
	$y_2 \geq 0$	(c)
	$y_1 \geq 0$	(d)
	$-y_1 + y_2 \leq 3$	(e)

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```

Objective: max $c^T x$

Subject to: $A x \leq b$

$x \geq 0$

Running Time (n = # variables, m = # constraints defined by A):

- Number of possible neighbors for a given vertex?

n non-negativity constraints are forming the vertex, we will replace one of them with one of the 'regular' constraints.

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

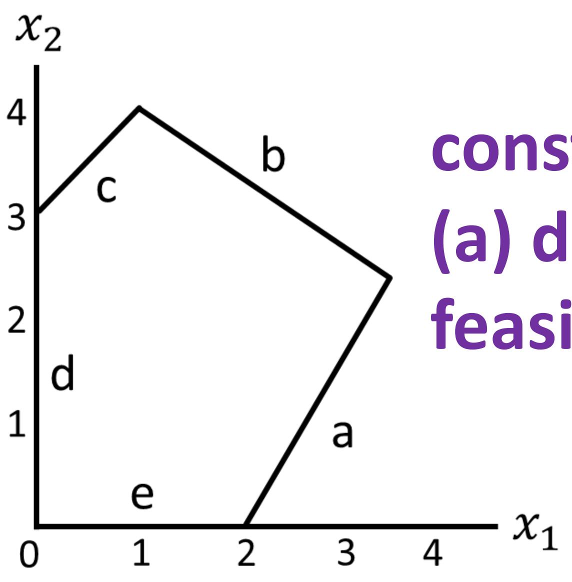
Objective: $\max c^T x$
Subject to: $A x \leq b$
$x \geq 0$

Running Time (n = # variables, m = # constraints defined by A):

- Number of possible neighbors for a given vertex?

n non-negativity constraints are forming the vertex, we will replace one of them with one of the 'regular' constraints. Thus, nm options for neighboring vertices

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```



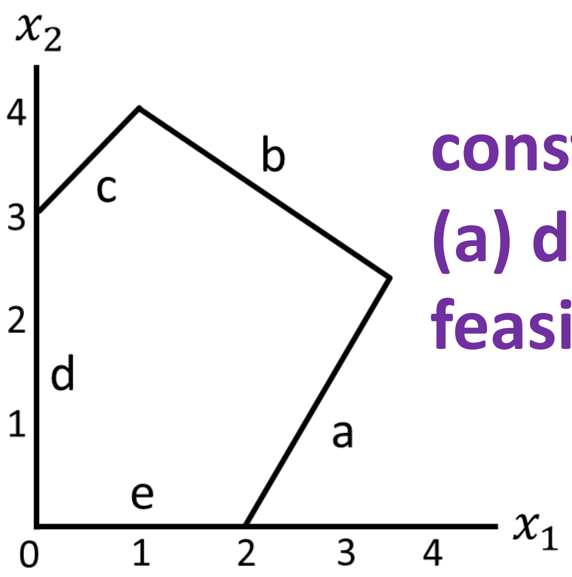
constraints (c) and (a) do not form a feasible vertex.

Running Time (n = # variables, m = # constraints defined by A):

- Number of possible neighbors for a given vertex?

n non-negativity constraints are forming the vertex, we will replace one of them with one of the 'regular' constraints. Thus, nm options for neighboring vertices, but not all combinations are feasible.

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```



constraints (c) and (a) do not form a feasible vertex.

Running Time (n = # variables, m = # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$

n non-negativity constraints are forming the vertex, we will replace one of them with one of the 'regular' constraints. Thus, nm options for neighboring vertices, but not all combinations are feasible.

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i - a_i·x
      reformulate LP in terms of y_i
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Running Time ($n =$ # variables, $m =$ # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$
- Testing if possible neighbor is feasible $\in O(n^3)$

```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Objective: max $c^T x$

Subject to: $A x \leq b$

$x \geq 0$

Running Time ($n =$ # variables, $m =$ # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$
- Testing if possible neighbor is feasible $\in O(n^3)$

$\Rightarrow O(n^4 m)$ per iteration

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Running Time (n = # variables, m = # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$
- Testing if possible neighbor is feasible $\in O(n^3)$

$\Rightarrow$ ~~$O(n^4 m)$ per iteration~~ **Actually $O(nm)$**

Since we don't consider all possible neighbors and swapping coordinate systems (rewriting the LP) can happen in $O((m+n)n)$.

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i – a_i·x
      reformulate LP in terms of y_i
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Running Time ($n =$ # variables, $m =$ # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$
- Testing if possible neighbor is feasible $\in O(n^3)$

  $\Rightarrow$ ~~$O(n^4 m)$ per iteration~~ Actually $O(nm)$
- Possible number of iterations?

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint a_i·x ≤ b_i
         y_i = b_i − a_i·x
      reformulate LP in terms of y_i
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Running Time ($n =$ # variables, $m =$ # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$
- Testing if possible neighbor is feasible $\in O(n^3)$

  $\Rightarrow$ ~~$O(n^4 m)$ per iteration~~ Actually $O(nm)$

- Possible number of iterations?

Possible number of feasible vertices

```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi = bi – ai·x
      reformulate LP in terms of yi
   return v
```

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

$m + n$ constraints in total.

Running Time (n = # variables, m = # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$
- Testing if possible neighbor is feasible $\in O(n^3)$

  $\Rightarrow$ ~~$O(n^4 m)$ per iteration~~ Actually $O(nm)$

- Possible number of iterations $= \binom{m+n}{n}$

Possible number of feasible vertices

```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi =
      reformu
   return v
```

Objective: $\max c^T x$

Subject to: $A x \leq b$

$x \geq 0$

nts

$$\binom{m+n}{n} \geq \binom{2n}{n} = \frac{(2n)!}{n!n!} = \frac{(2n)(2n-1)(2n-2)(2n-3)(2n-4)\ldots}{n(n-1)(n-2)\ldots n(n-1)(n-2)\ldots}$$

$$= \frac{2\cancel{n}(2n-1)2\cancel{(n-1)}(2n-3)2\cancel{(n-2)}\ldots}{\cancel{n(n-1)(n-2)\ldots}n(n-1)(n-2)\ldots}$$

$$\geq \frac{2^n(2n-1)(2n-3)\ldots}{n!} \geq 2^n$$

Runnin

-
- Testing if possible neighbor is feasible $\in O(n^3)$

  $\Rightarrow$ ~~$O(n^4m)$ per iteration~~ Actually $O(nm)$
- Possible number of iterations $= \binom{m+n}{n}$

Possible number of feasible vertices

```
Simplified_Simplex(LP)
   v = origin
   while cj > 0 for some j
      relax tight constraint.
      stop at new constraint.
      v = new intersection.
      for constraint ai·x ≤ bi
         yi =
      reformu
   return v
```

Objective: $\max c^T x$

Subject to: $A x \le b$

$x \ge 0$

nts

$$\binom{m+n}{n} \ge \binom{2n}{n} = \frac{(2n)!}{n!n!} = \frac{(2n)(2n-1)(2n-2)(2n-3)(2n-4)\ldots}{n(n-1)(n-2)\ldots n(n-1)(n-2)\ldots}$$

$$= \frac{2\cancel{n}(2n-1)2\cancel{(n-1)}(2n-3)2\cancel{(n-2)}\ldots}{\cancel{n(n-1)(n-2)\ldots}n(n-1)(n-2)\ldots}$$

$$\ge \frac{2^n(2n-1)(2n-3)\ldots}{n!} \ge 2^n$$

Runnir

- Testing if possible neighbor is feasible $\in O(n^3)$

  $\Rightarrow$ ~~$O(n^4 m)$ per iteration~~ Actually $O(nm)$

- Possible number of iterations $= \binom{m+n}{n} \in \Omega(2^n)$

Possible number of feasible vertices

```
Simplified_Simplex(LP)
   v = origin
   while c_j > 0 for some j
      relax tight constraint.
      stop at new constrai
      v = new int
      for cor
         y_i =
      reformu ate LP in terms of y_i
   return v
```

Obj

max $c^T x$

$A x \leq b$

$x \geq 0$

Simplex is Exponential!

Running Time (n = # variables, m = # constraints defined by A):

- Number of possible neighbors for a given vertex $\in O(nm)$
- Testing if possible neighbor is feasible $\in O(n^3)$

  $\Rightarrow$ ~~$O(n^4 m)$ per iteration~~ Actually $O(nm)$
- Possible number of iterations $= \binom{m+n}{n} \in \Omega(2^n)$

```
Simplified_Simplex(LP)
    v = origin
    while c_j > 0 for some j
        relax tight constraint.
        stop at new constrai
        v = new int
        for con
            y_i =
        reformu   LP in terms of y_i
```

Obj

$$\max c^T x$$
$$A x \le b$$
$$x \ge 0$$

Simplex is Exponential!

History Lesson:

- Solving systems of linear inequalities dates back to the 1800's.
- Linear programming was widely studied in the 1940's.
- Simplex invented in 1947.
- Specific LPs that Simplex takes exponential time on discovered in 1972.
- Soviet mathematician found polynomial time algorithm in 1979.
- Interior point method found in 1984.
- CPLEX released in 1988.

Simplex – Starting Vertex

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

The origin is not always feasible!

Simplex – Starting Vertex

Objective:	$\max c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Objective:	$\min z_1 + \cdots + z_m$
Subject to:	$A x + z = b$
	$x \geq 0$
	$z \geq 0$

Ensure $b \geq 0$. (Negate equality if not)

Simplex – Starting Vertex

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

$\Rightarrow$

Objective:	$\min z_1 + \cdots + z_m$
Subject to:	$A x + z = b$
	$x \geq 0$
	$z \geq 0$

Simple feasible solution to new LP?

Simplex – Starting Vertex

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Objective:	$\min z_1 + \cdots + z_m$
Subject to:	$A x + z = b$
	$x \geq 0$
	$z \geq 0$

Simple feasible solution to new LP?

$z = b, x = 0$

What if optimal is 0?

Simplex – Starting Vertex

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

⇒

Objective:	$\min z_1 + \cdots + z_m$
Subject to:	$A x + z = b$
	$x \geq 0$
	$z \geq 0$

Simple feasible solution to new LP?

$z = b, x = 0$

What if optimal is 0?

$z_i = 0$ for each i, and whatever the selected x_i's are, are feasible.

What if optimal is > 0?

Simplex – Starting Vertex

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Objective:	$\min z_1 + \cdots + z_m$
Subject to:	$A x + z = b$
	$x \geq 0$
	$z \geq 0$

Simple feasible solution to new LP?

$z = b$, $x = 0$

What if optimal is 0?

$z_i = 0$ for each i, and whatever the selected x_i's are, are feasible.

What if optimal is > 0?

$A x + 0 = b$ is not feasible, which means **original LP is not feasible!!**

Simplex – Unbounded Solution

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

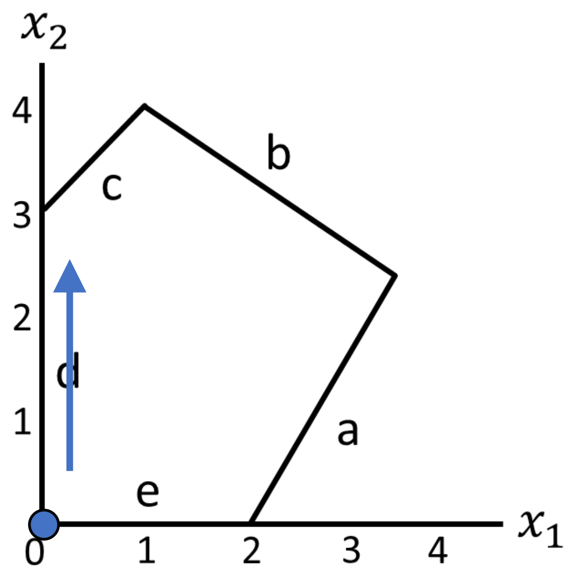
How will we know if the solution is unbounded?

Simplex – Unbounded Solution

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

How will we know if the solution is unbounded?

Simplex will attempt to relax a constraint, and it never meets another bounding constraint.



Simplex – Degenerate Vertices

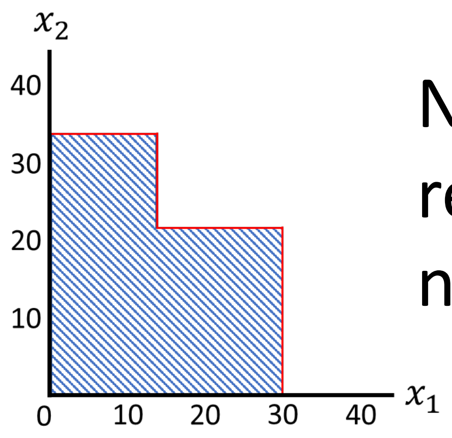
Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Can we ever have the same (non-optimal) objective value on neighboring vertices?

Simplex – Degenerate Vertices

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Can we ever have the same (non-optimal) objective value on neighboring vertices?

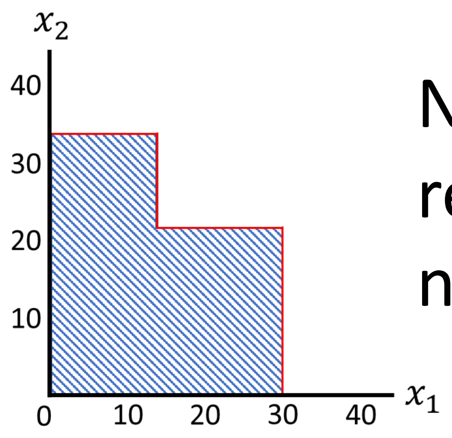


No. Feasible region would not be convex.

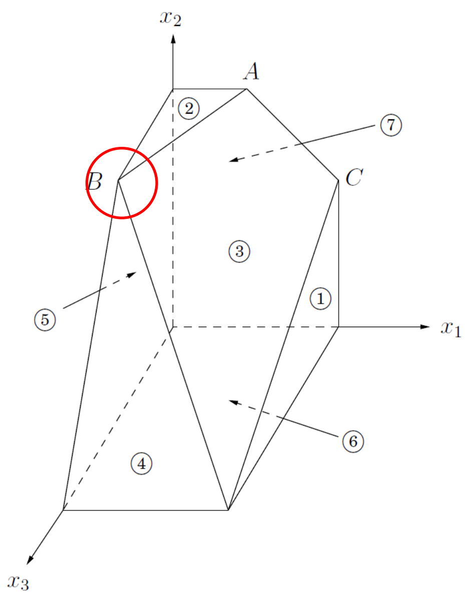
Simplex – Degenerate Vertices

Objective:	max $c^T x$
Subject to:	$A x \leq b$
	$x \geq 0$

Can we ever have the same (non-optimal) objective value on neighboring vertices?



No. Feasible region would not be convex.



Yes. Vertex in $\mathbb{R}^n$ can be intersection of $> n$ constraints.