Algorithms and Data Structures

Introduction to Linear Programming

A problem

A problem

A company makes two products, X and Y, using two machines, A and B.

A problem

A company makes two products, X and Y, using two machines, A and B.

Each unit of product X requires *50min* processing time on machine A and *30min* processing time on machine B.

A problem

A company makes two products, X and Y, using two machines, A and B.

Each unit of product X requires *50min* processing time on machine A and *30min* processing time on machine B.

Each unit of product Y requires *24min* of processing time on machine A and *33min* of processing time on machine B.

A problem

A company makes two products, X and Y, using two machines, A and B.

Each unit of product X requires *50min* processing time on machine A and *30min* processing time on machine B.

Each unit of product Y requires *24min* of processing time on machine A and *33min* of processing time on machine B.

At the start of the week, there are *30 units* of X and *90 units* of Y in stock.

A problem

A company makes two products, X and Y, using two machines, A and B.

Each unit of product X requires *50min* processing time on machine A and *30min* processing time on machine B.

Each unit of product Y requires *24min* of processing time on machine A and *33min* of processing time on machine B.

At the start of the week, there are *30 units* of X and *90 units* of Y in stock.

The available processing time on machine A is *40 hours* and on machine B it is *35 hours*.

A problem

A company makes two products, X and Y, using two machines, A and B.

Each unit of product X requires *50min* processing time on machine A and *30min* processing time on machine B.

Each unit of product Y requires *24min* of processing time on machine A and *33min* of processing time on machine B.

At the start of the week, there are *30 units* of X and *90 units* of Y in stock.

The available processing time on machine A is *40 hours* and on machine B it is *35 hours*.

The demand for X in the week is *75 units* and for Y it is *95 units*.

A problem

A company makes two products, X and Y, using two machines, A and B.

Each unit of product X requires *50min* processing time on machine A and *30min* processing time on machine B.

Each unit of product Y requires *24min* of processing time on machine A and *33min* of processing time on machine B.

At the start of the week, there are *30 units* of X and *90 units* of Y in stock.

The available processing time on machine A is *40 hours* and on machine B it is *35 hours*.

The demand for X in the week is *75 units* and for Y it is *95 units*.

Goal: Maximise the combined sum of units of X and Y in stock at the end of the week.

A linear program

Goal: Maximise the combined sum of units of X and Y in stock at the end of the week.

Maximise $(x + 30 - 75) + (y + 90 - 95)$

A linear program

Each unit of product X requires *50min* processing time on machine A and *30min* processing time on machine B.

Each unit of product Y requires *24min* of processing time on machine A and *33min* of processing time on machine B.

At the start of the week, there are *30 units* of X and *90 units* of Y in stock.

The available processing time on machine A is *40 hours* and on machine B it is *35 hours*.

The demand for X in the week is *75 units* and for Y it is *95 units*.

$$50x + 24y \leq 2400 \qquad x \geq 75 - 30$$

$$30x + 33y \leq 2100 \qquad y \geq 95 - 90$$

A linear program

Maximise $x + y - 50$

subject to

$$50x + 24y \leq 2400$$
$$30x + 33y \leq 2100$$
$$x \geq 45$$
$$y \geq 5$$

Linear programming (LP)

$$\text{maximise} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1\in}^{n} \alpha_{ij} x_j \leq b_i, \quad i = 1, ..., m$$

$$x_j \geq 0, \quad j = 1, ..., n$$

Linear programming (in matrix form)

$$\begin{aligned} \text{maximise} \quad & c^{\mathrm{T}} x \\ \text{subject to} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned}$$

Terminology

Terminology

Solution: An assignment of values to the variables.

Terminology

Solution: An assignment of values to the variables.

Feasible solution: A solution that satisfies all of the constraints.

Terminology

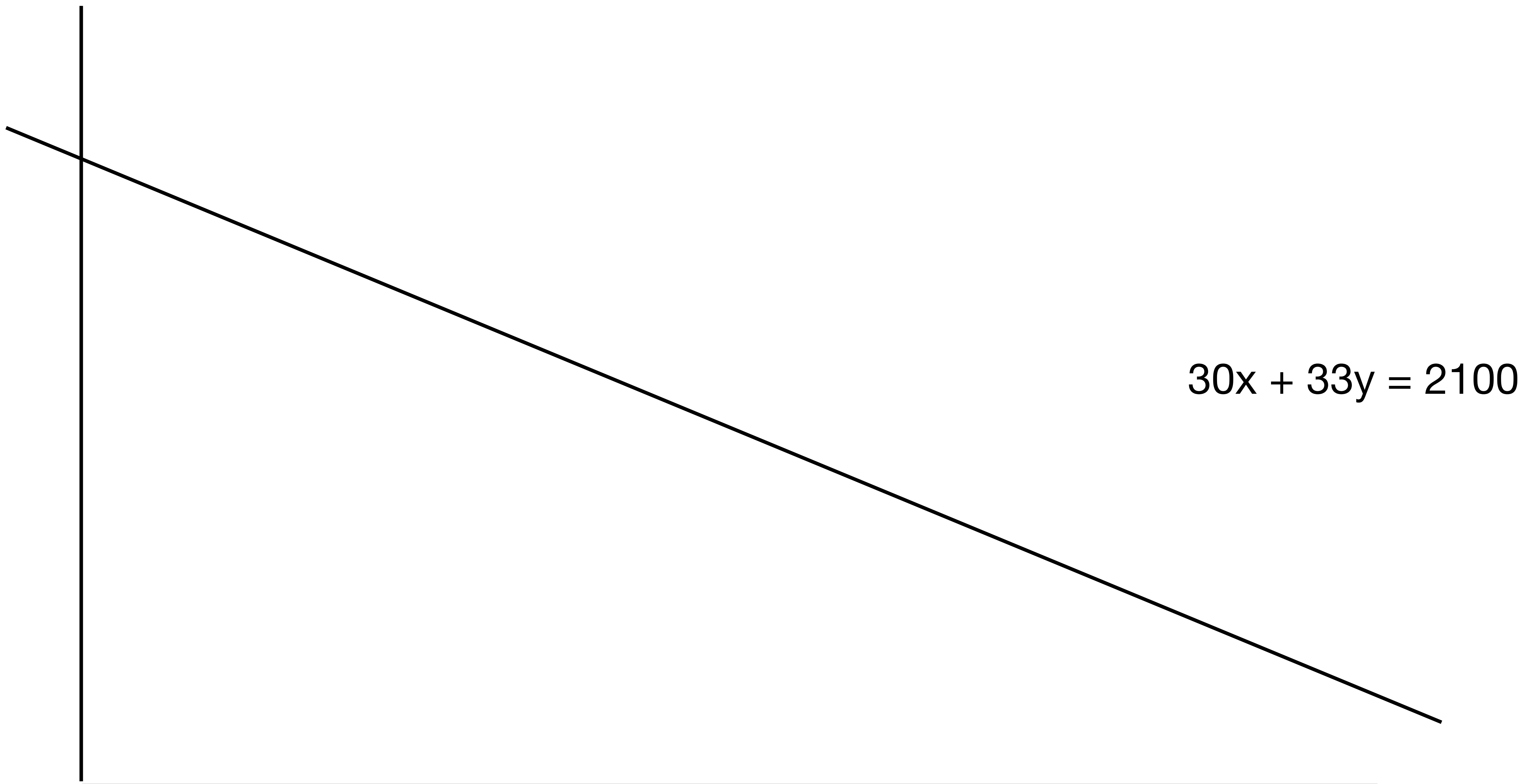
Solution: An assignment of values to the variables.

Feasible solution: A solution that satisfies all of the constraints.

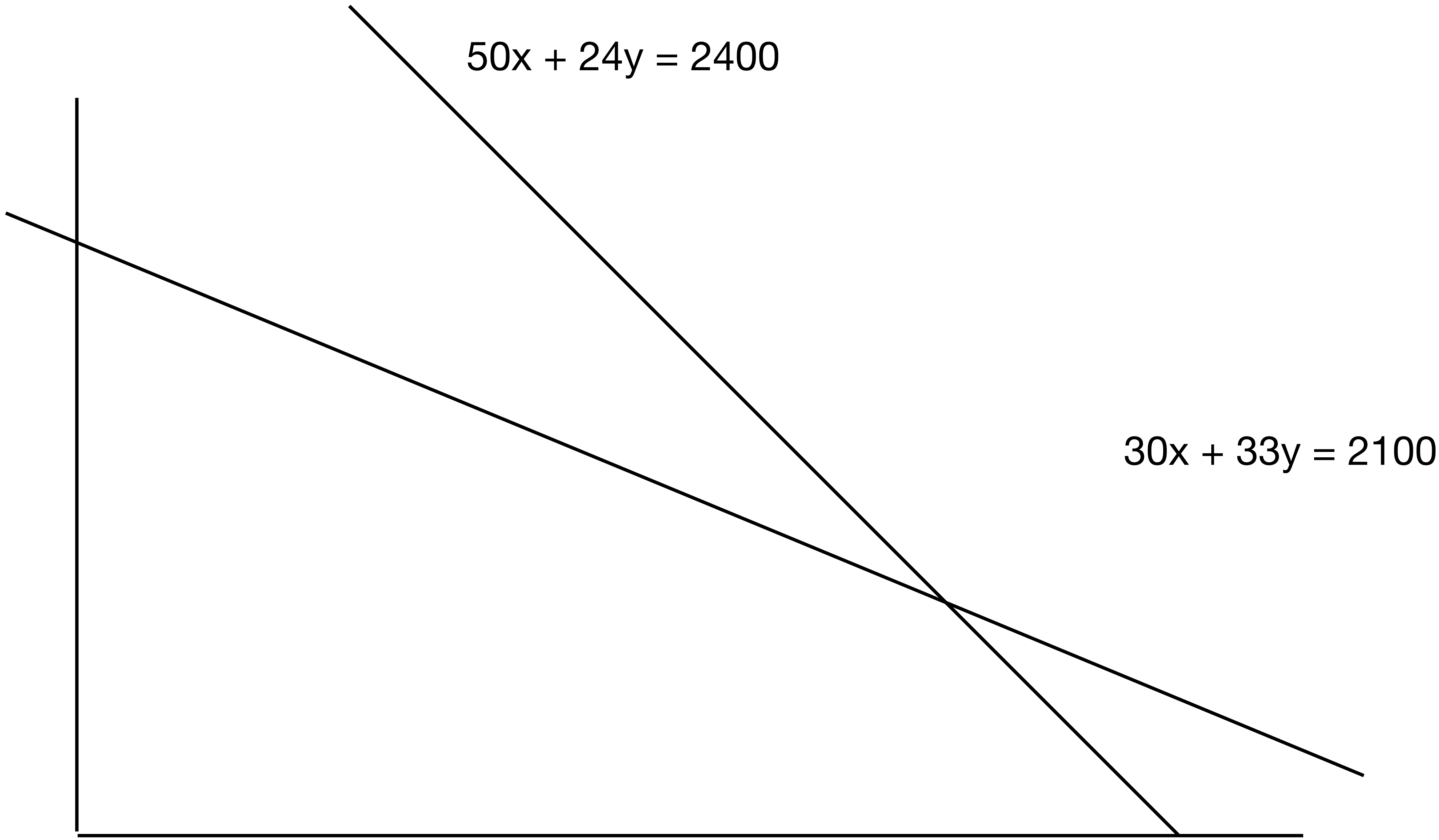
Feasible region: The set of feasible solutions.

Geometric Interpretation

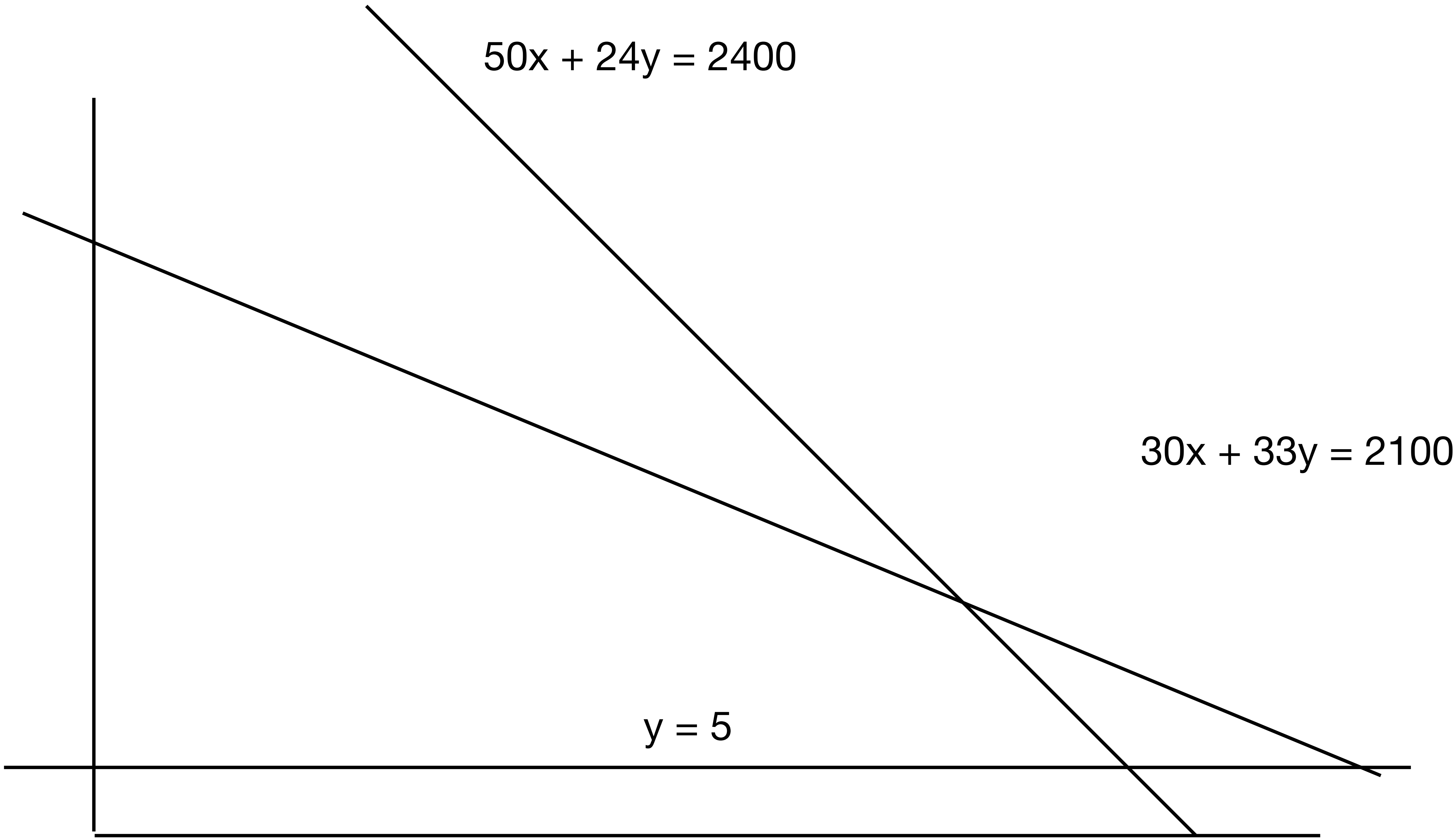
Geometric Interpretation



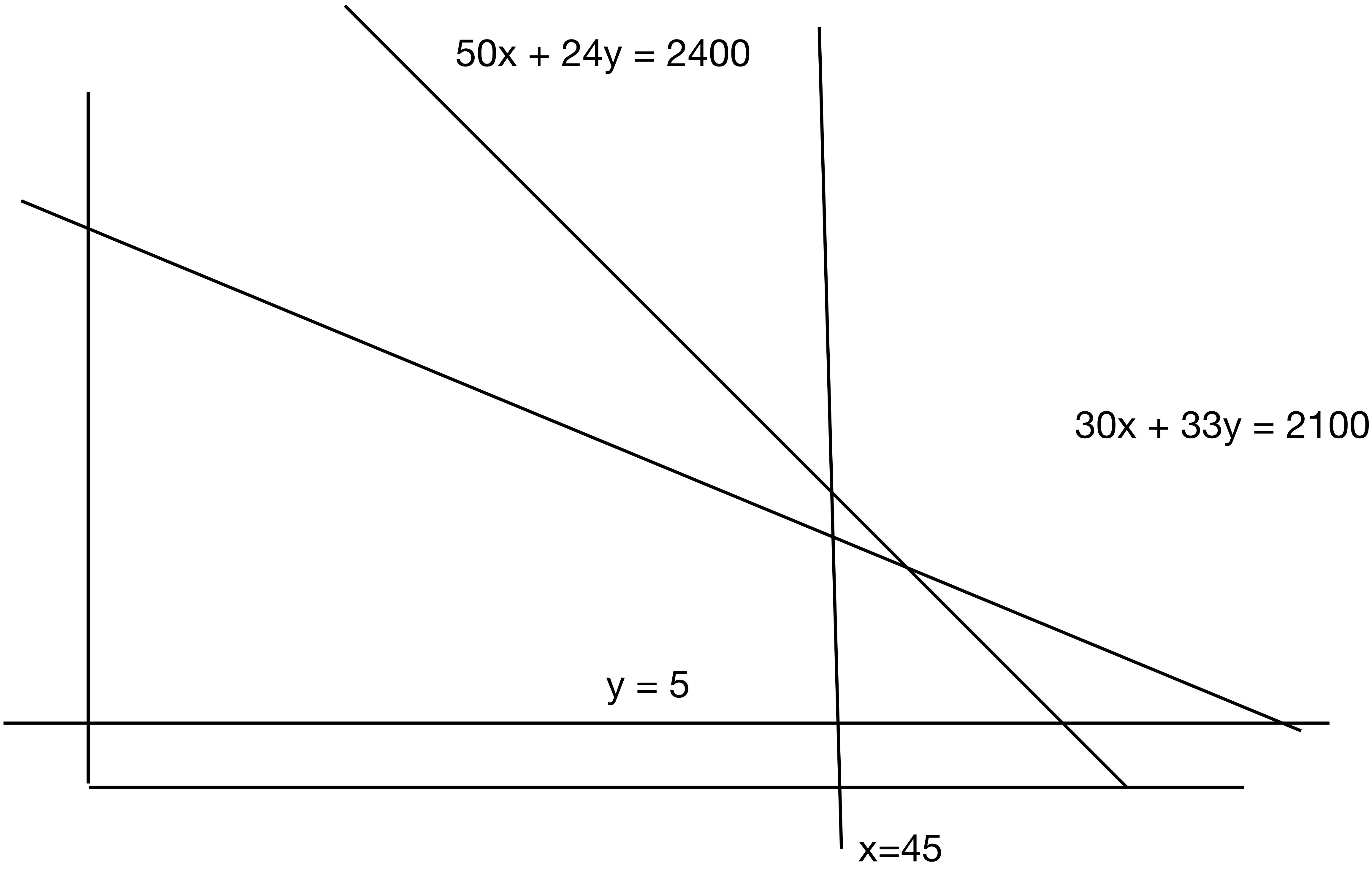
Geometric Interpretation



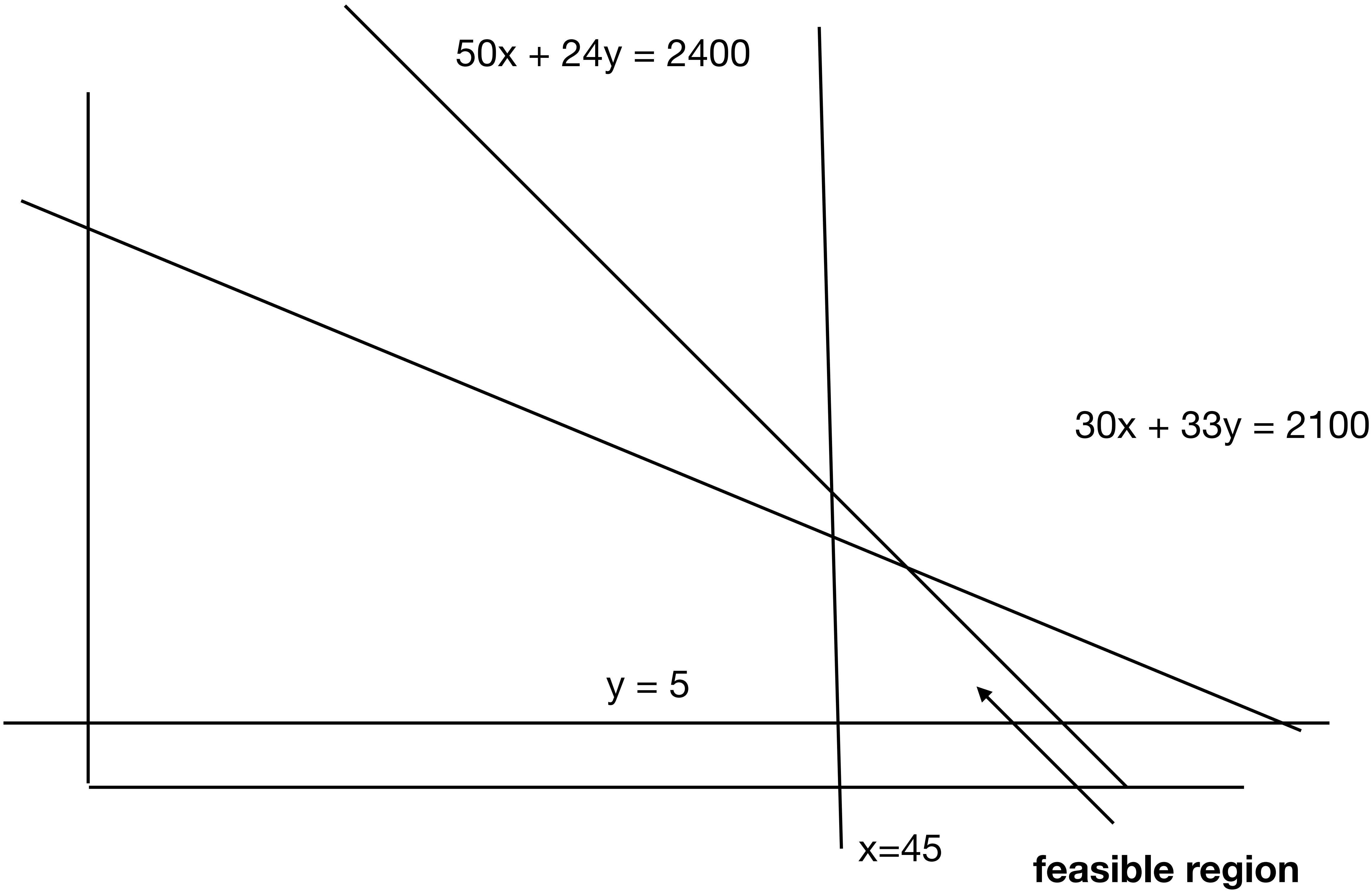
Geometric Interpretation



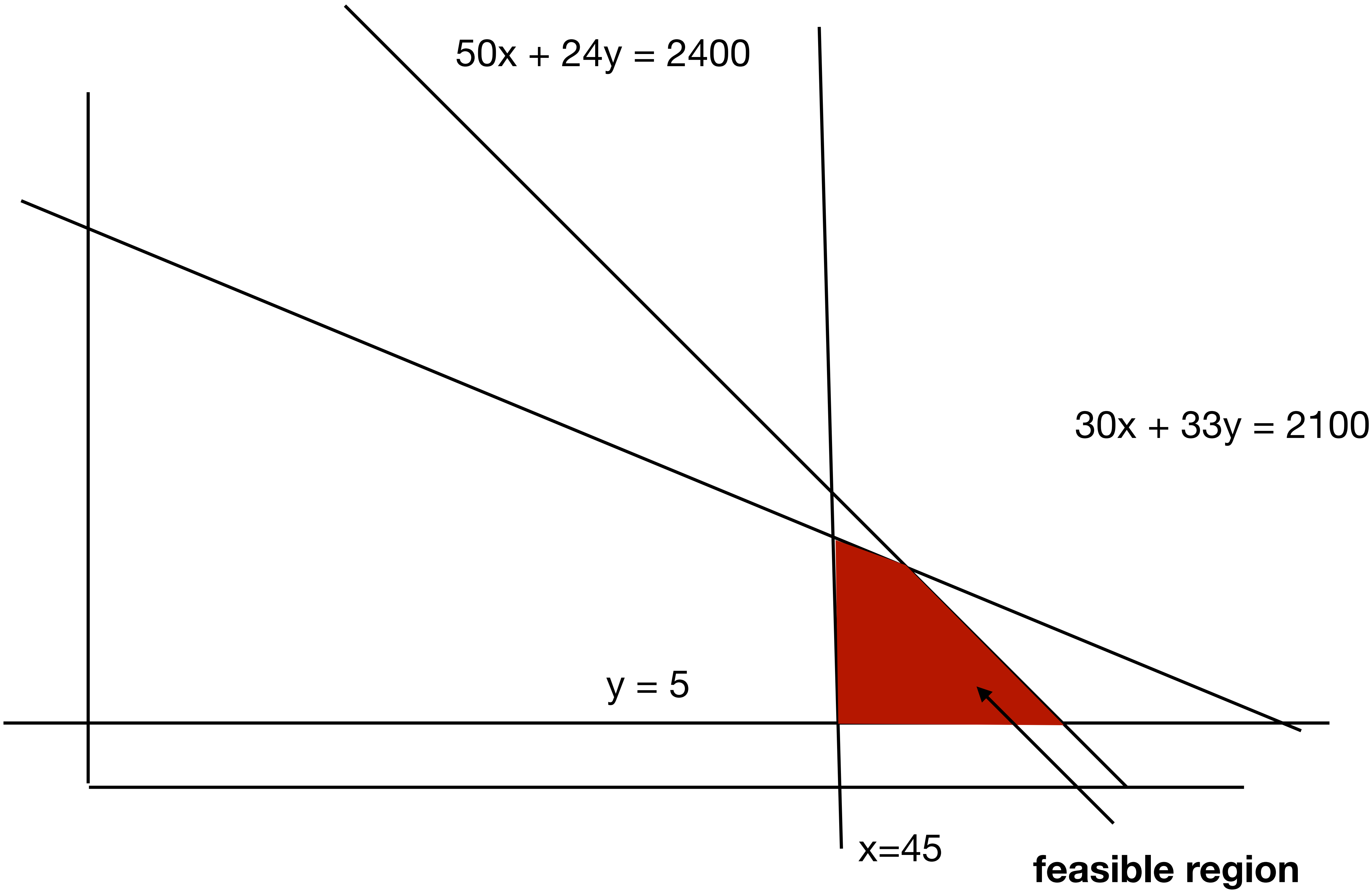
Geometric Interpretation



Geometric Interpretation



Geometric Interpretation



Do all LPs have feasible solutions?

Maximise $5x + 4y$

subject to
$$x + y \leq 2$$
$$-2x - 2y \leq -9$$
$$x, y \geq 0$$

Do all LPs have feasible solutions?

Maximise $5x + 4y$

subject to
$$\boxed{\begin{aligned} x + y &\leq 2 \\ -2x - 2y &\leq -9 \end{aligned}}$$
one contradicts the other!
$$x, y \geq 0$$

Terminology

Solution: An assignment of values to the variables.

Feasible solution: A solution that satisfies all of the constraints.

Feasible region: The set of feasible solutions.

Terminology

Solution: An assignment of values to the variables.

Feasible solution: A solution that satisfies all of the constraints.

Feasible region: The set of feasible solutions.

An LP that does not have any feasible solutions is called infeasible.

Terminology

Solution: An assignment of values to the variables.

Feasible solution: A solution that satisfies all of the constraints.

Feasible region: The set of feasible solutions.

An LP that does not have any feasible solutions is called infeasible.

Optimal solution: A feasible solution with the maximum possible value for the objective function.

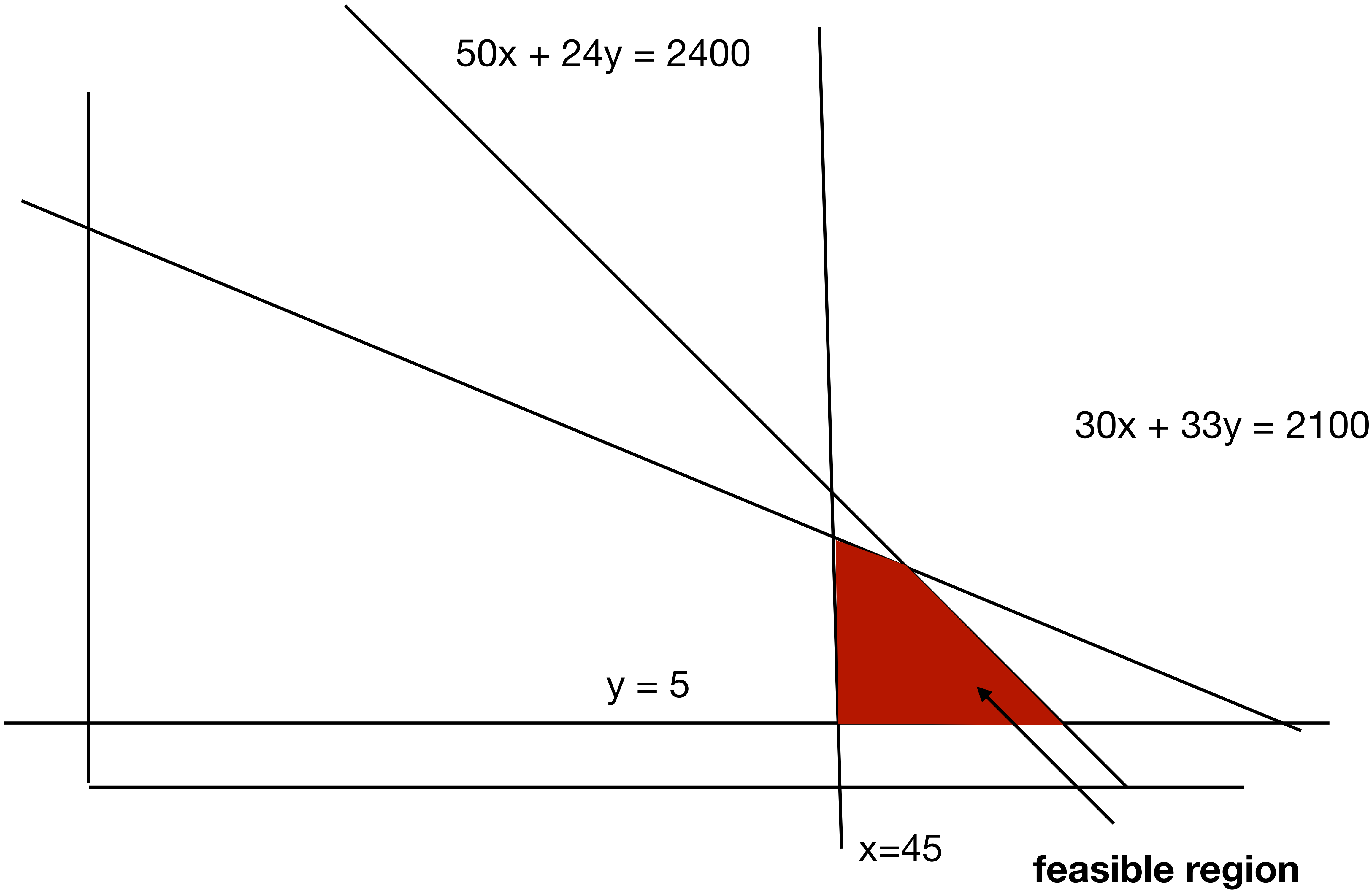
Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

e.g., $50x+24y - 2400 = x - 45$

Solving the linear program



Diet Example

Diet Example

Assume that we have two energy X and Y which provide calories, vitamin A and vitamin C daily.

Diet Example

Assume that we have two energy X and Y which provide calories, vitamin A and vitamin C daily.

We would like to drink x bottles of X and y bottles of Y, to ensure that our daily intake is at least 300 calories, 36 units of vitamin A and 90 units of vitamin C.

Diet Example

Assume that we have two energy X and Y which provide calories, vitamin A and vitamin C daily.

We would like to drink x bottles of X and y bottles of Y, to ensure that our daily intake is at least 300 calories, 36 units of vitamin A and 90 units of vitamin C.

One bottle of X provides 60 calories, 12 units of vitamin A, and 10 units of vitamin C.

Diet Example

Assume that we have two energy X and Y which provide calories, vitamin A and vitamin C daily.

We would like to drink x bottles of X and y bottles of Y, to ensure that our daily intake is at least 300 calories, 36 units of vitamin A and 90 units of vitamin C.

One bottle of X provides 60 calories, 12 units of vitamin A, and 10 units of vitamin C.

One bottle of Y provides 60 calories, 6 units of vitamin A, and 30 units of vitamin C.

Diet Example

Assume that we have two energy X and Y which provide calories, vitamin A and vitamin C daily.

We would like to drink x bottles of X and y bottles of Y, to ensure that our daily intake is at least 300 calories, 36 units of vitamin A and 90 units of vitamin C.

One bottle of X provides 60 calories, 12 units of vitamin A, and 10 units of vitamin C.

One bottle of Y provides 60 calories, 6 units of vitamin A, and 30 units of vitamin C.

One bottle of X costs £12, whereas one bottle of Y costs £15.

Diet Example

Assume that we have two energy X and Y which provide calories, vitamin A and vitamin C daily.

We would like to drink x bottles of X and y bottles of Y, to ensure that our daily intake is at least 300 calories, 36 units of vitamin A and 90 units of vitamin C.

One bottle of X provides 60 calories, 12 units of vitamin A, and 10 units of vitamin C.

One bottle of Y provides 60 calories, 6 units of vitamin A, and 30 units of vitamin C.

One bottle of X costs £12, whereas one bottle of Y costs £15.

How do we maintain our diet goals at the lowest possible cost?

Diet Example

Minimise $12x + 15y$

subject to

$$60x + 60y \geq 300$$
$$12x + 6y \geq 36$$
$$10x + 30y \geq 90$$
$$x, y \geq 0$$

Diet Example

Minimise $12x + 15y$

subject to

$$x + y \geq 5$$

$$2x + y \geq 6$$

$$x + 3y \geq 9$$

$$x, y \geq 0$$

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

$x + y - 5 = 2x + y - 6 \Rightarrow x = 1$ and $y = 4$

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

$$x + y - 5 = 2x + y - 6 \Rightarrow x = 1 \text{ and } y = 4$$

$$12x + 15y = 72$$

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

$x + y - 5 = x + 3y - 9 \Rightarrow y = 2$ and $y = 3$

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

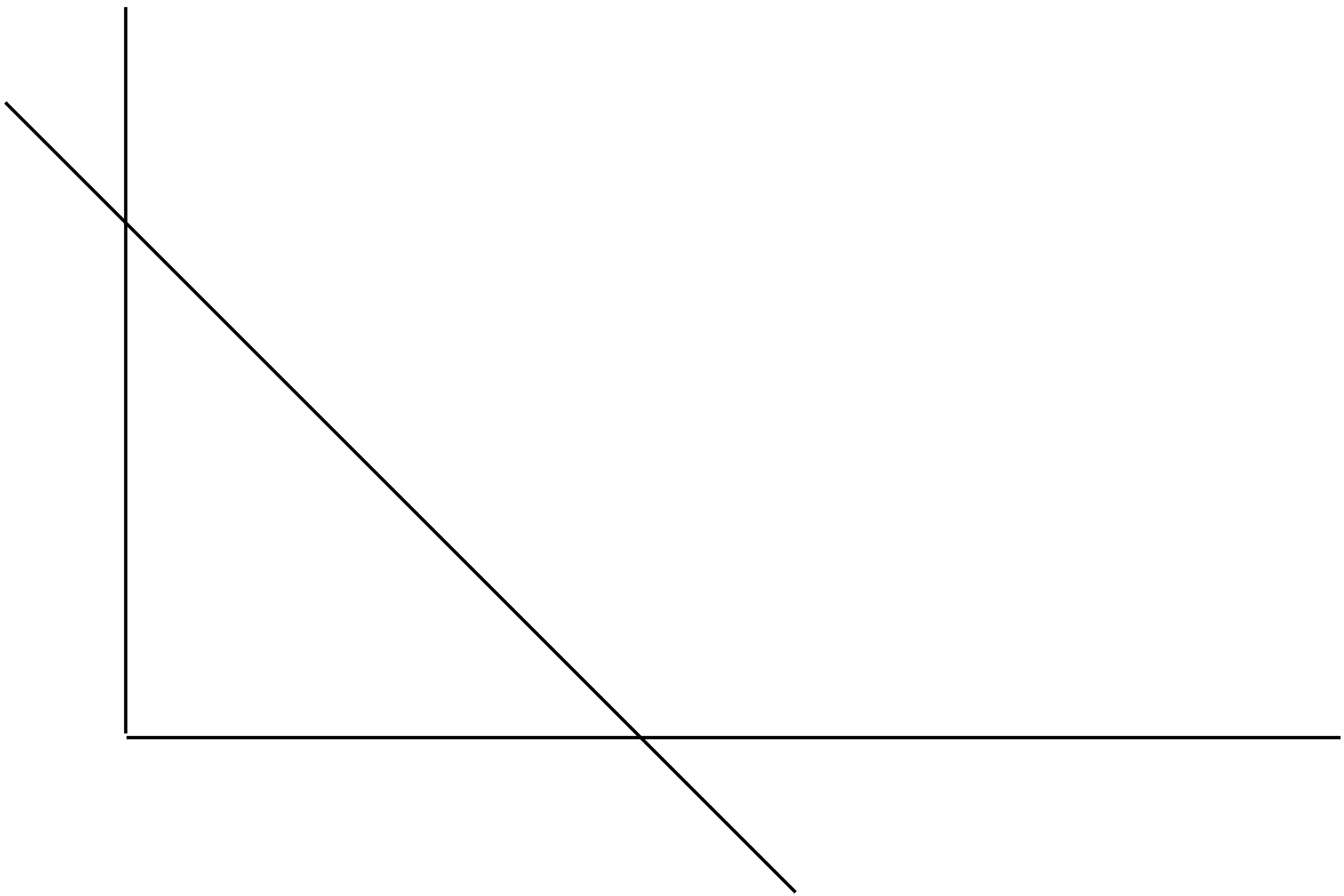
These are the intersection points of the lines defined by the constraints.

$$x + y - 5 = x + 3y - 9 \Rightarrow y = 2 \text{ and } y = 3$$

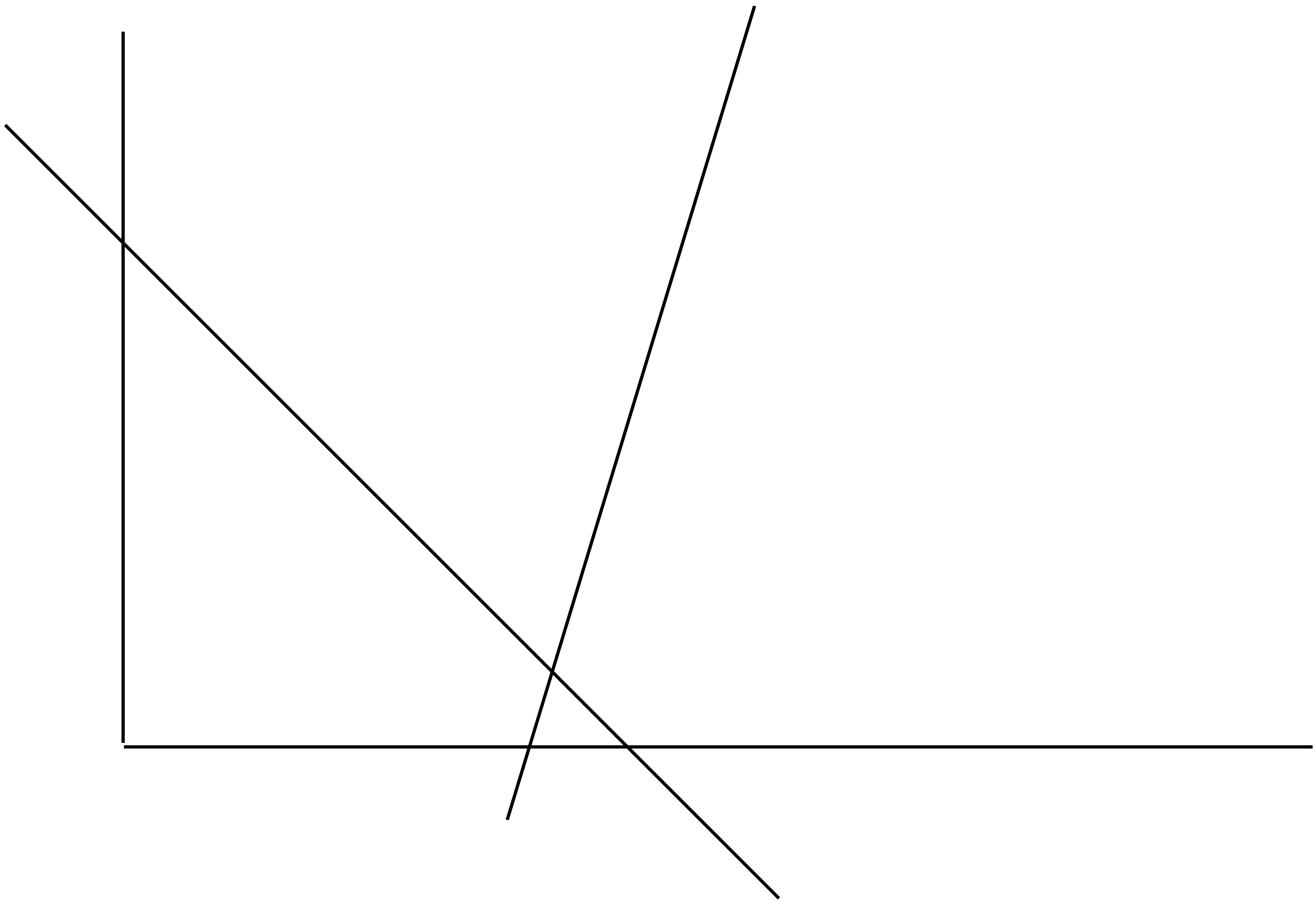
$$12x + 15y = 66$$

Geometric Interpretation

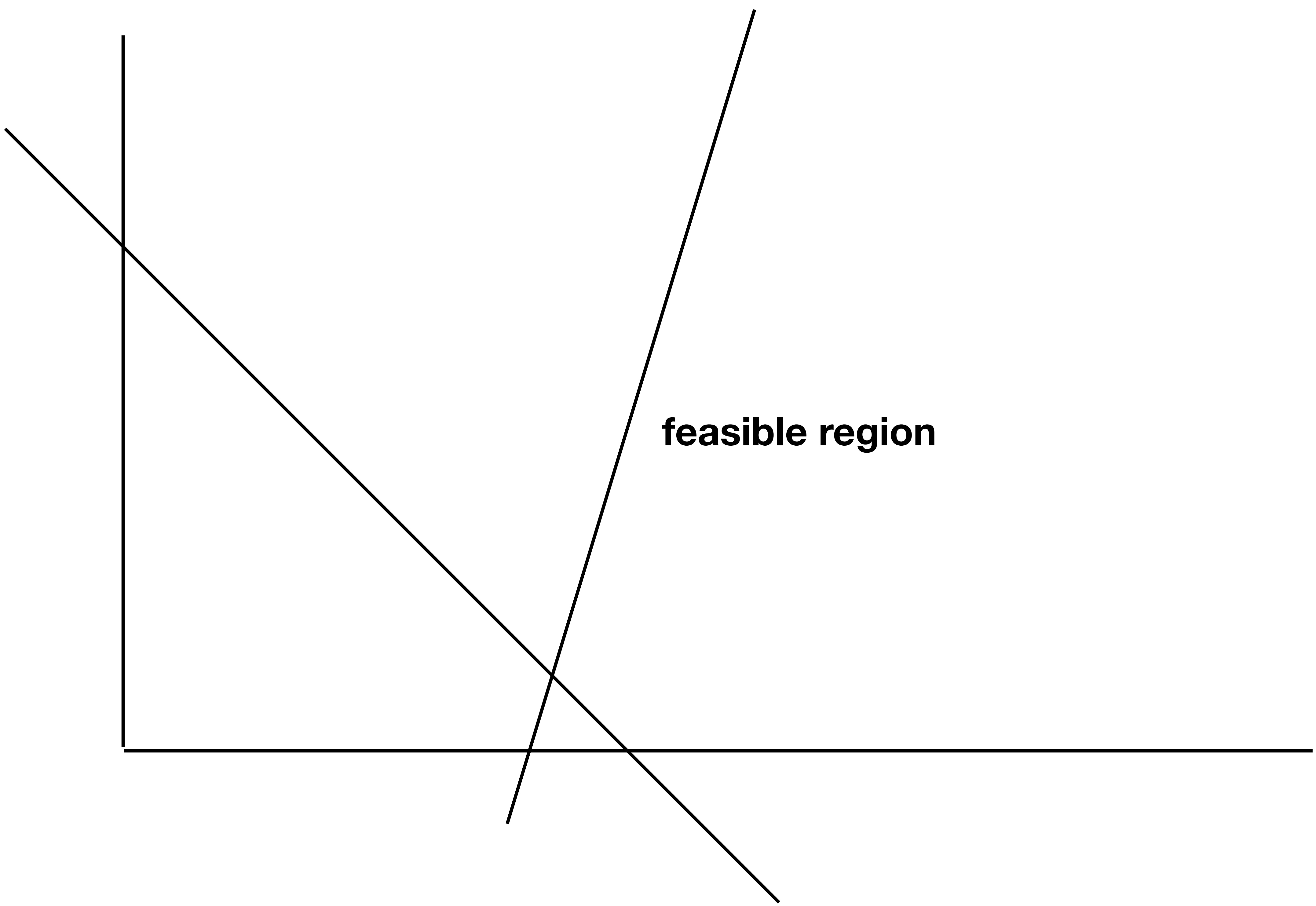
Geometric Interpretation



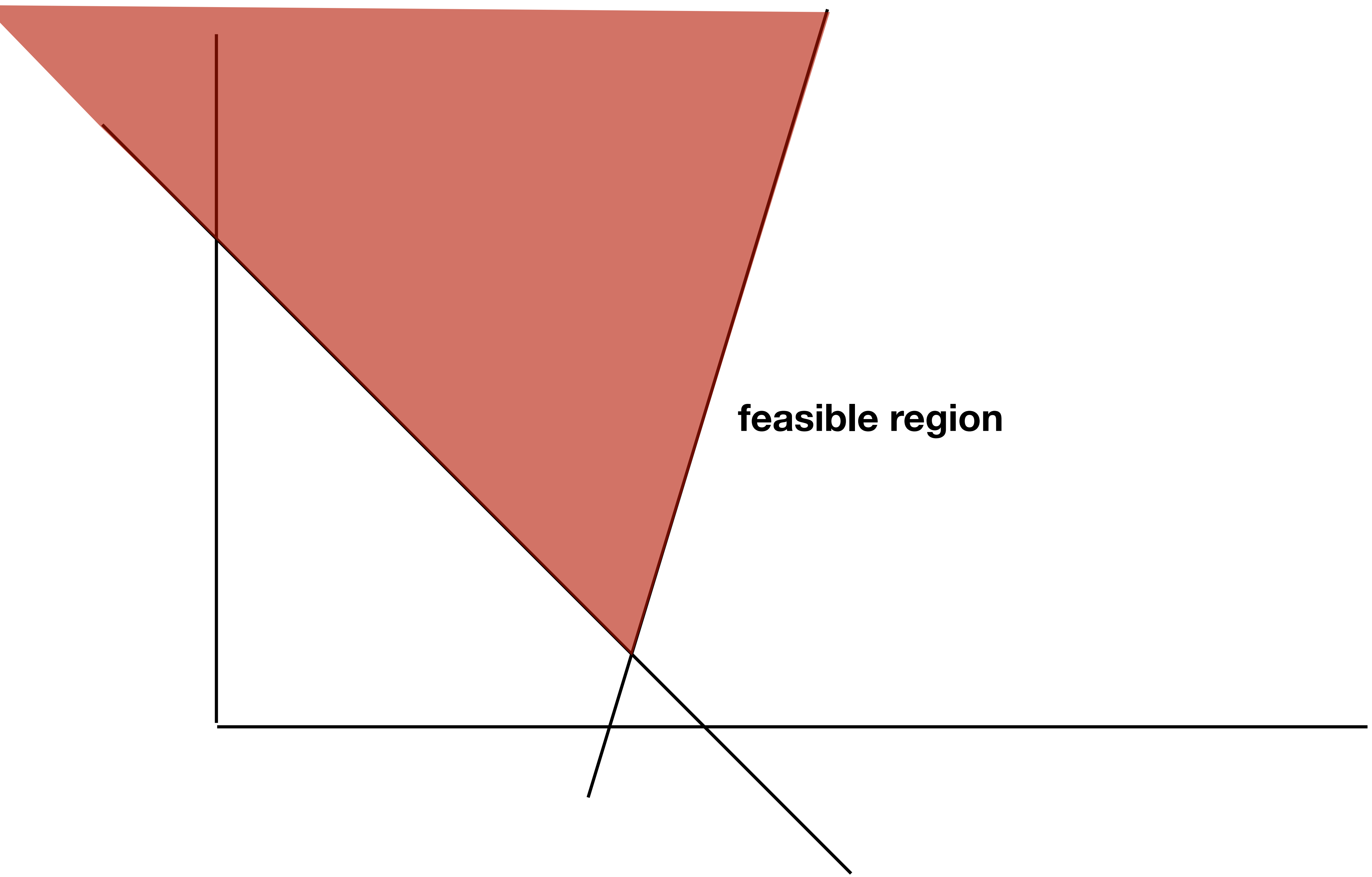
Geometric Interpretation



Geometric Interpretation



Geometric Interpretation



Terminology

Solution: An assignment of values to the variables.

Feasible solution: A solution that satisfies all of the constraints.

Feasible region: The set of feasible solutions.

An LP that does not have any feasible solutions is called infeasible.

Optimal solution: A feasible solution with the maximum possible value for the objective function

Terminology

Solution: An assignment of values to the variables.

Feasible solution: A solution that satisfies all of the constraints.

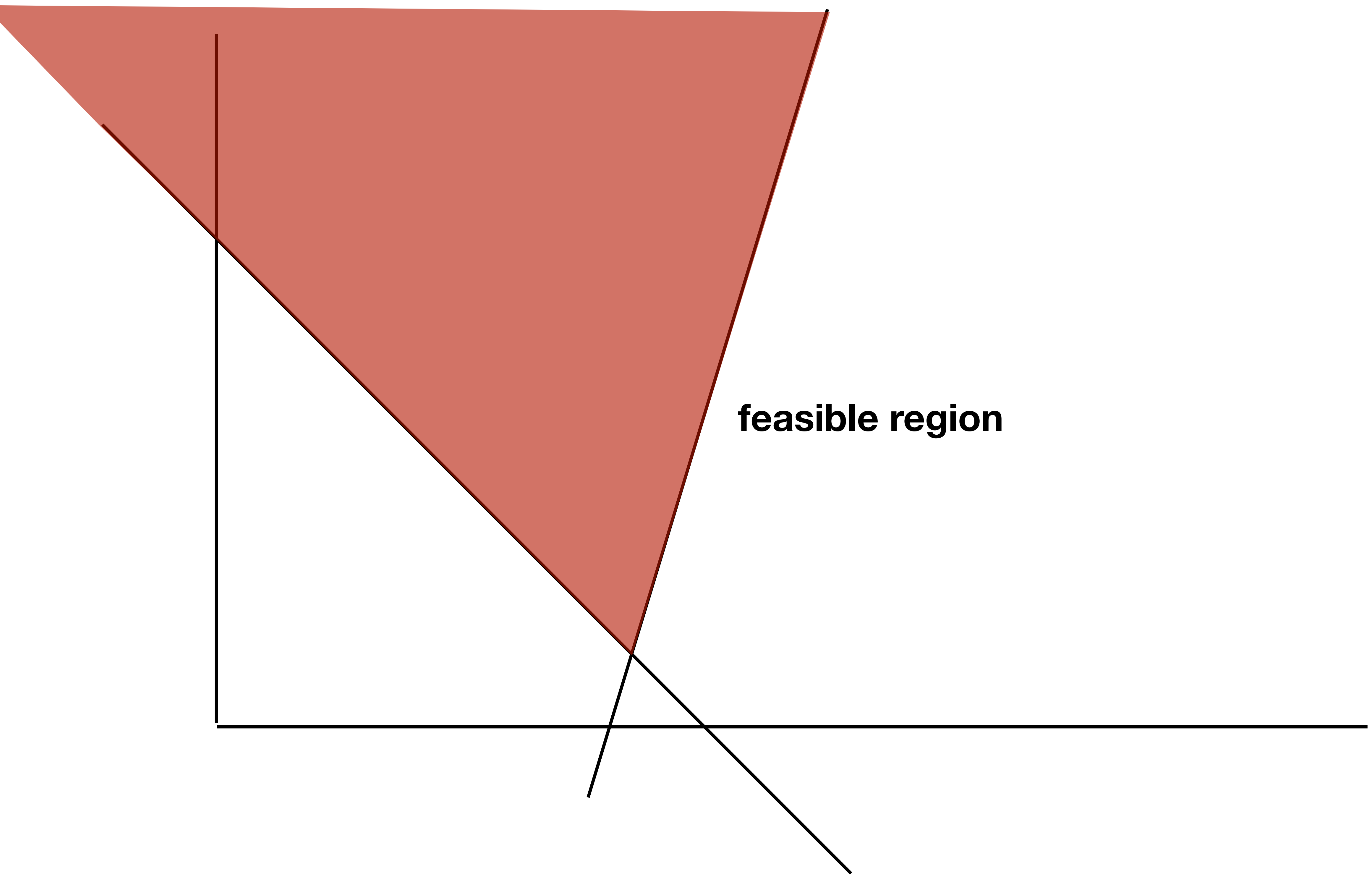
Feasible region: The set of feasible solutions.

An LP that does not have any feasible solutions is called infeasible.

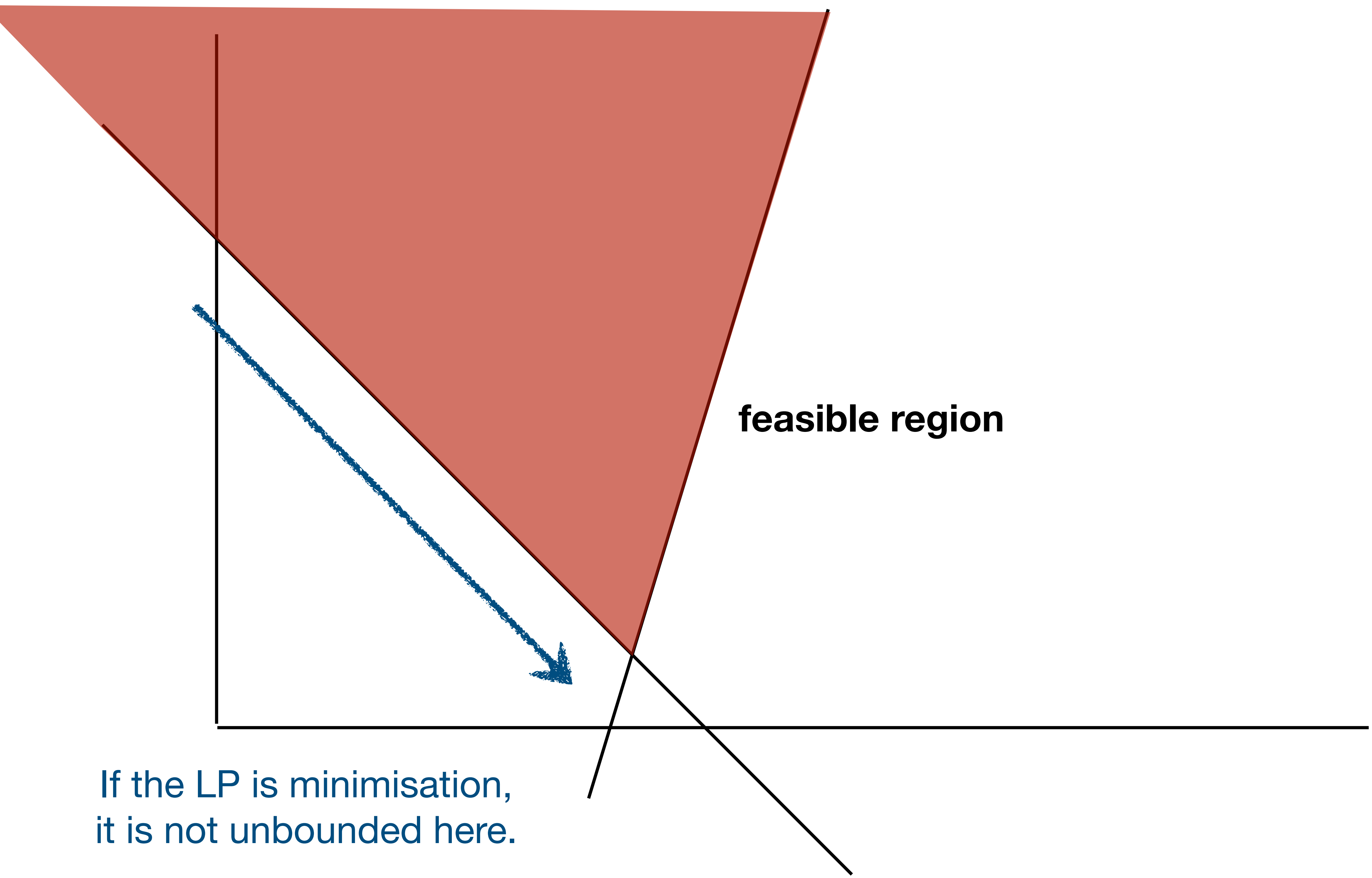
Optimal solution: A feasible solution with the maximum possible value for the objective function

An LP is called unbounded if it has feasible solutions with arbitrarily large objective values.

Unbounded LPs



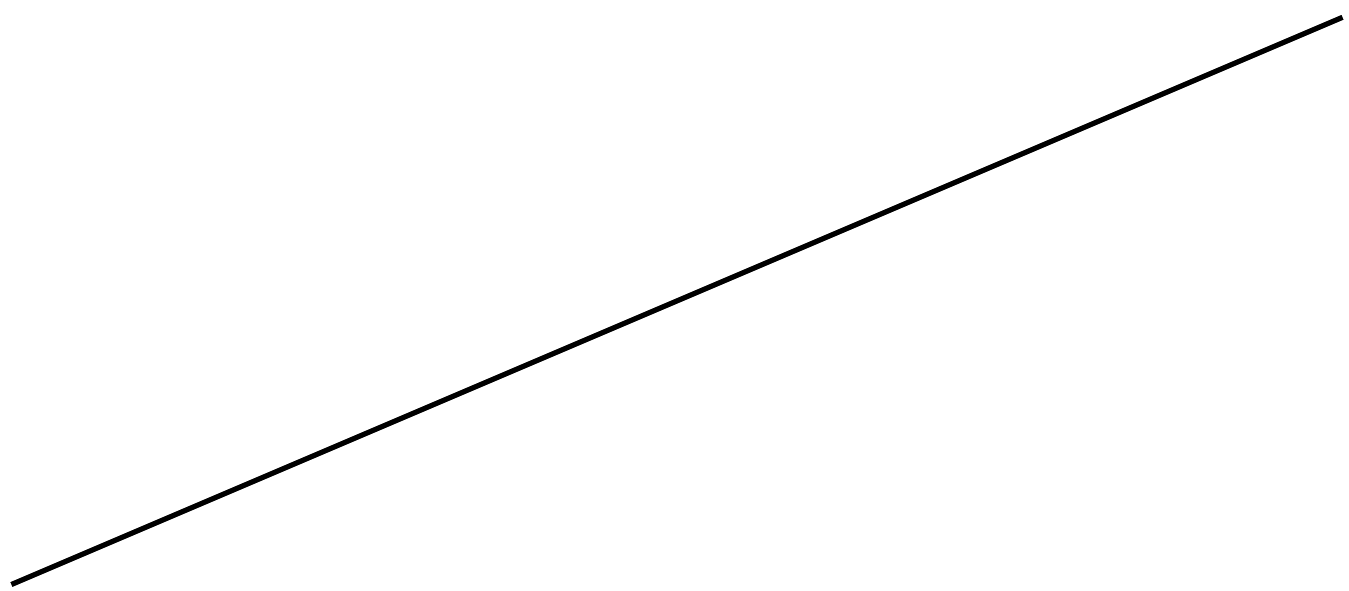
Unbounded LPs



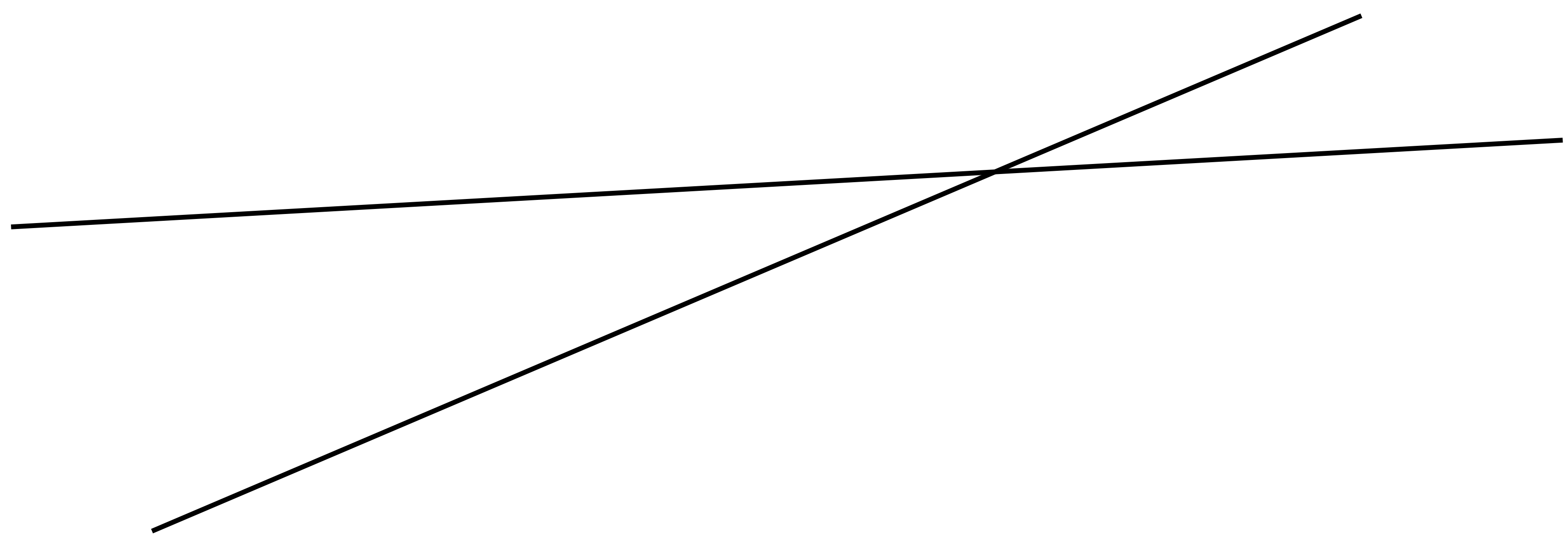
If the LP is minimisation,
it is not unbounded here.

More terminology

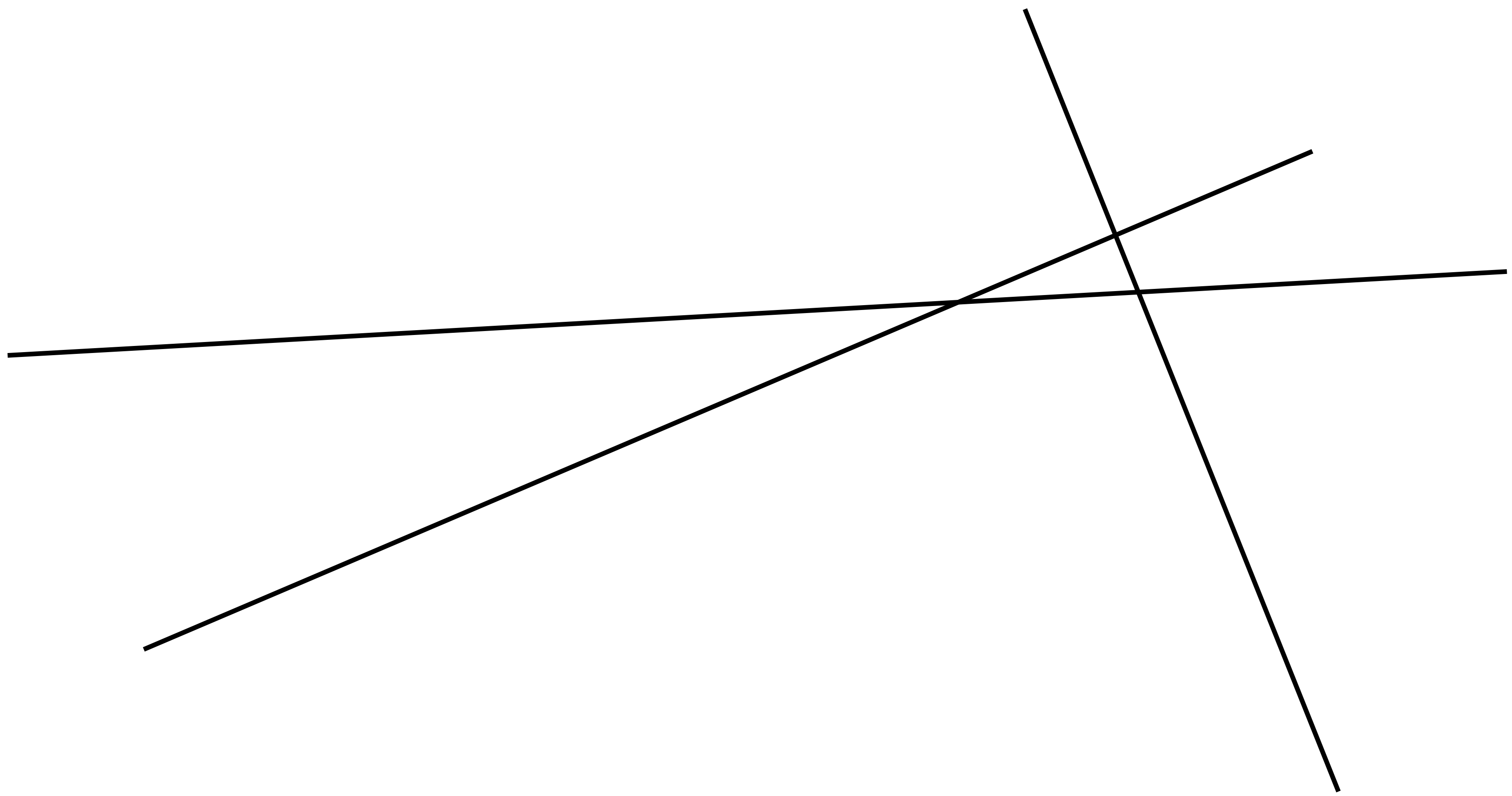
More terminology



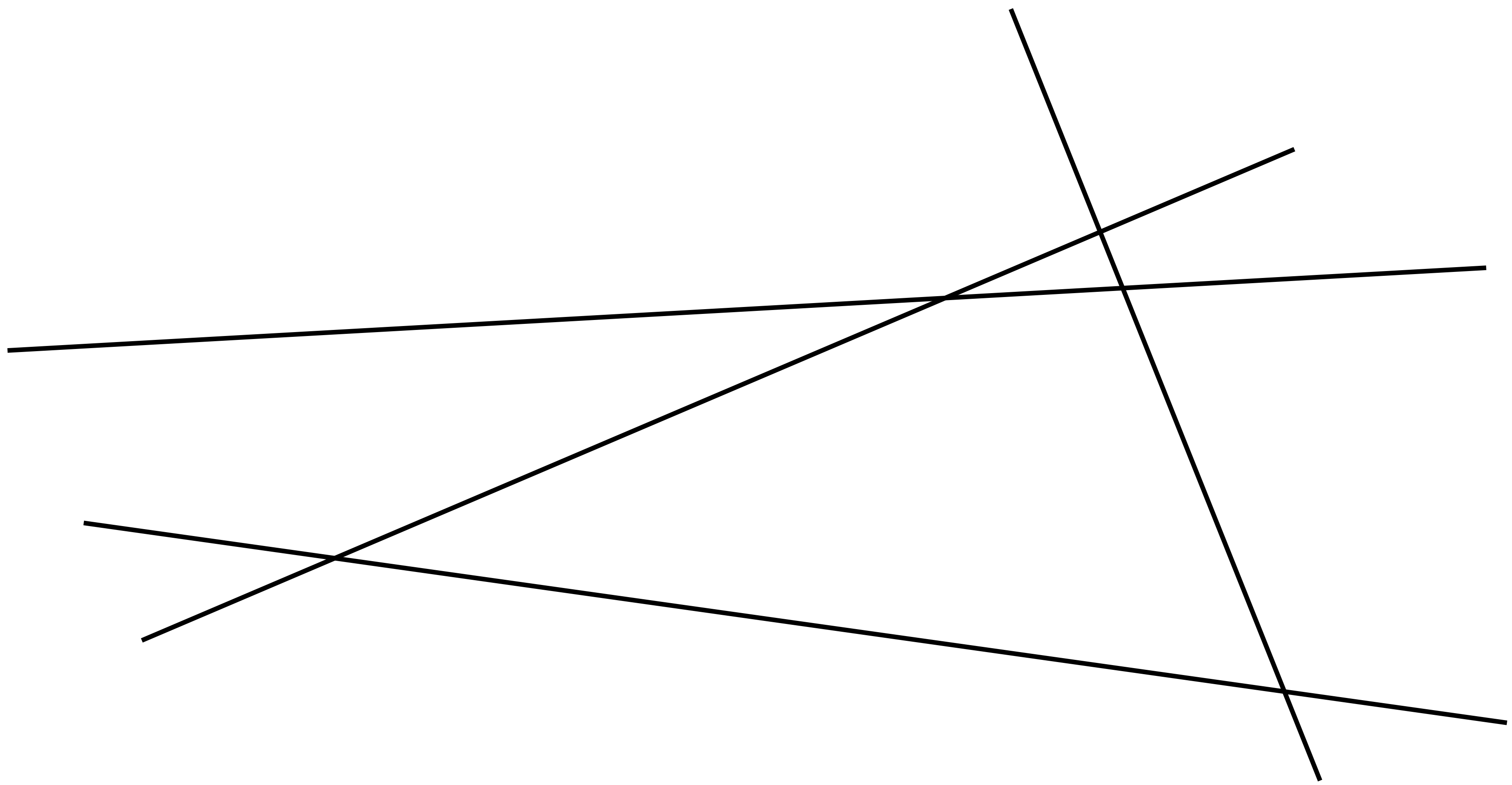
More terminology



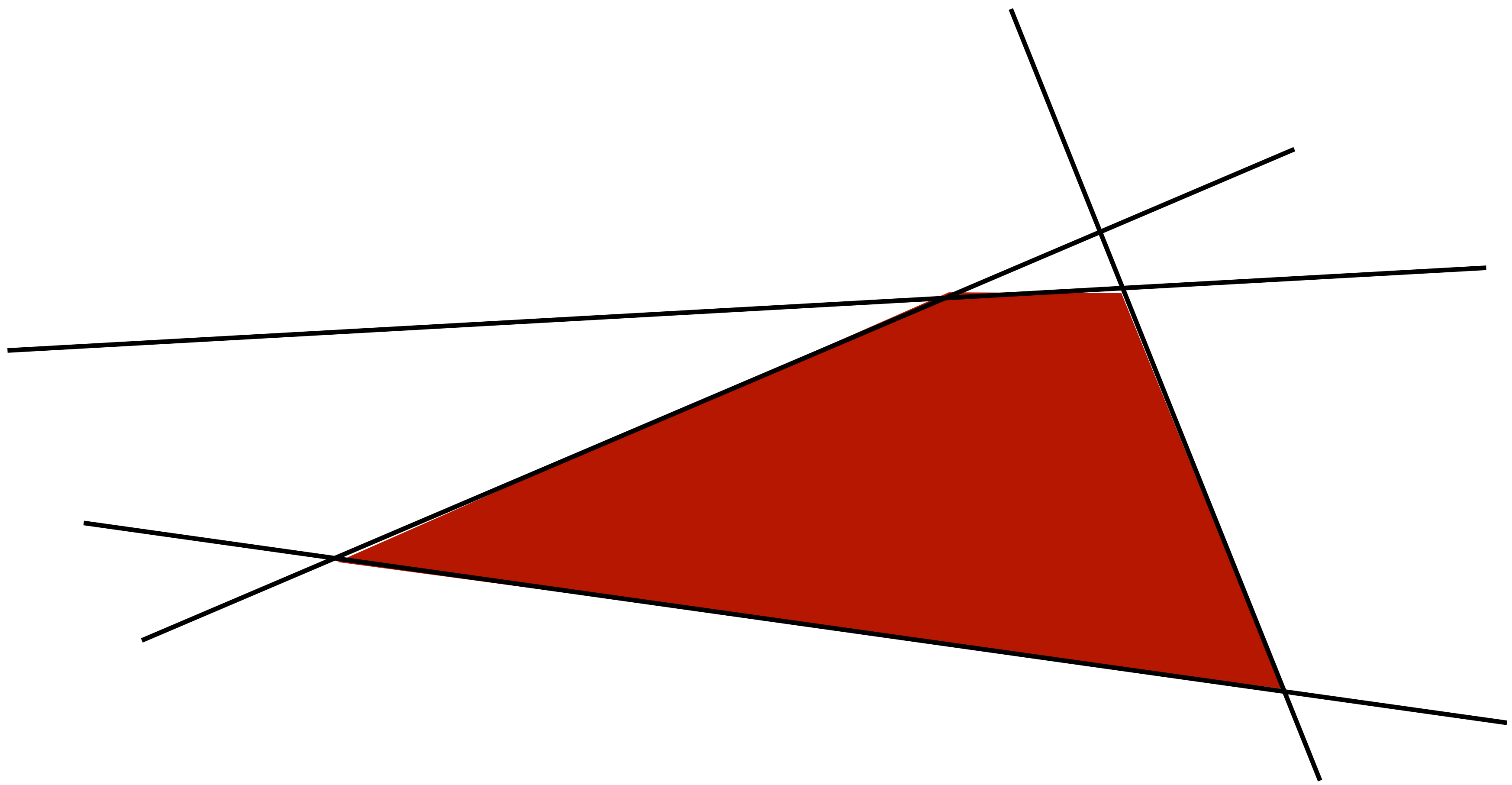
More terminology



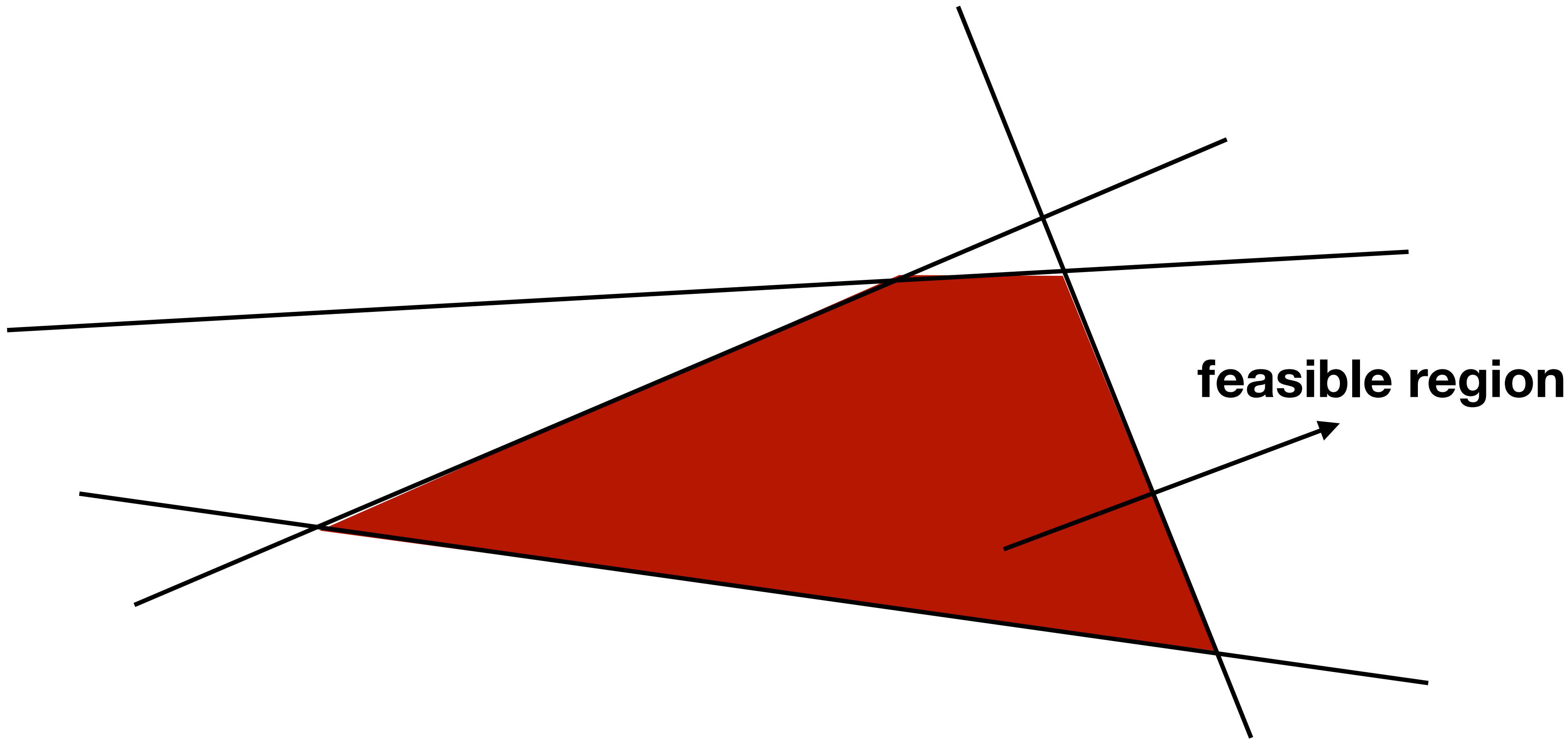
More terminology



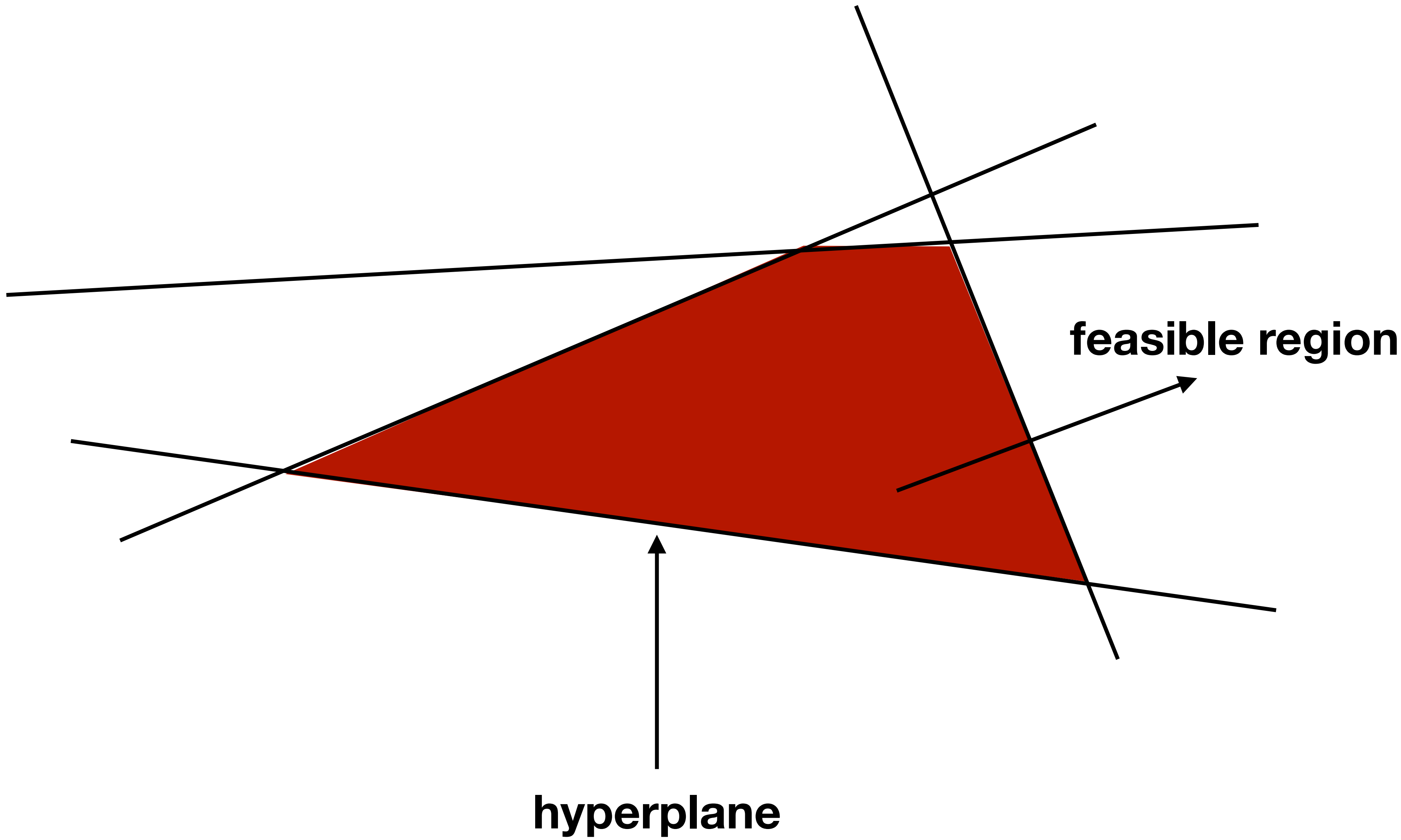
More terminology



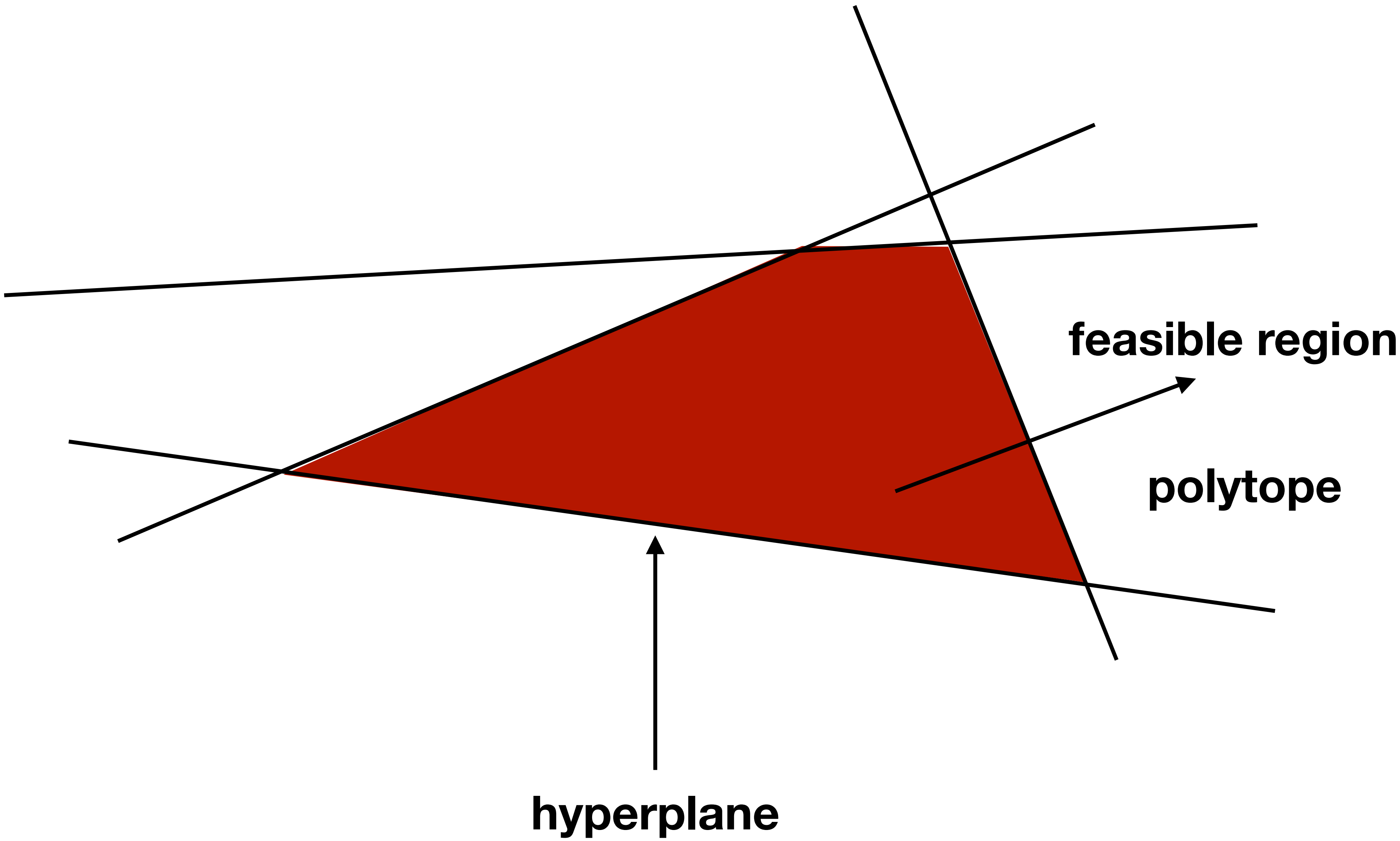
More terminology



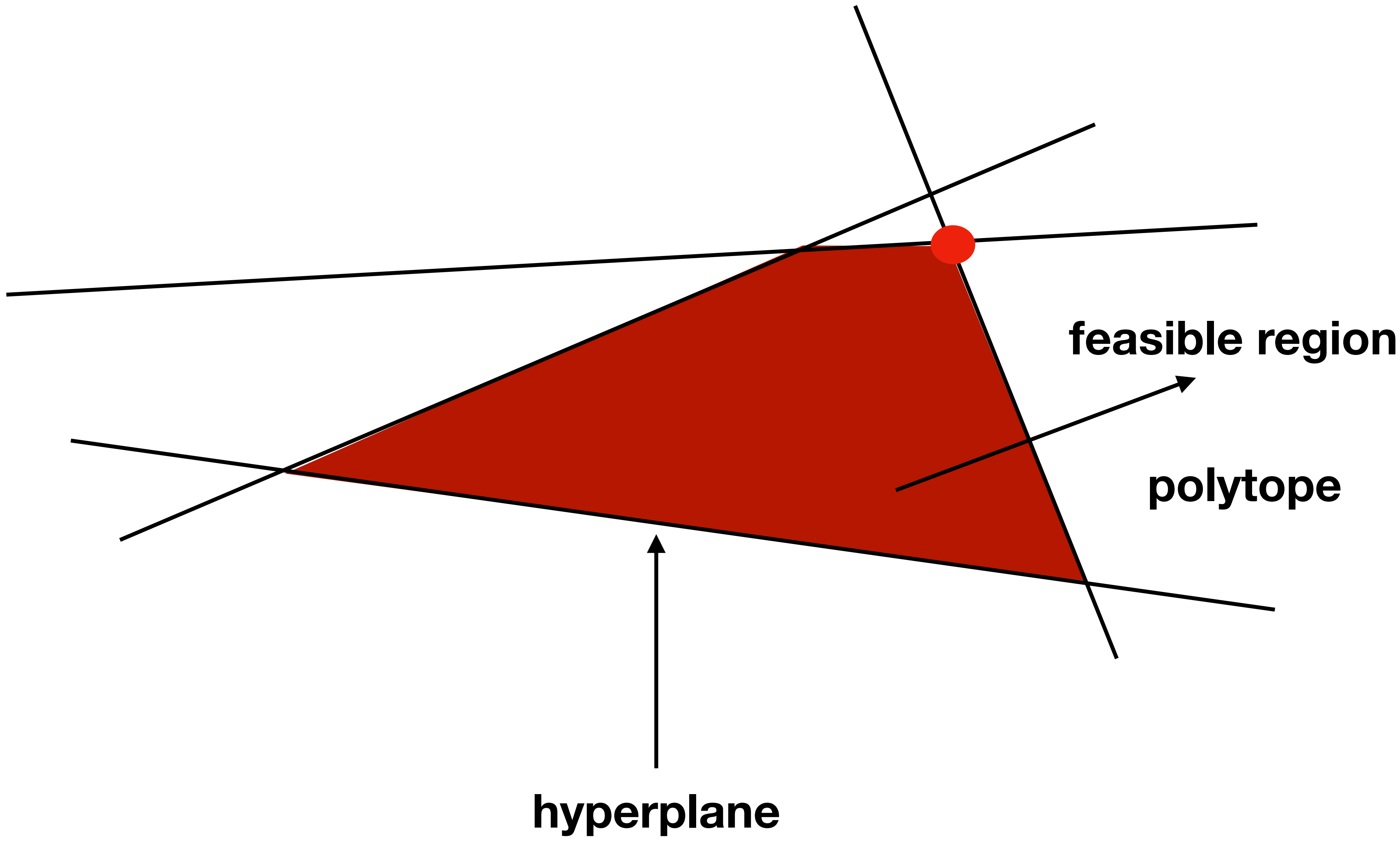
More terminology



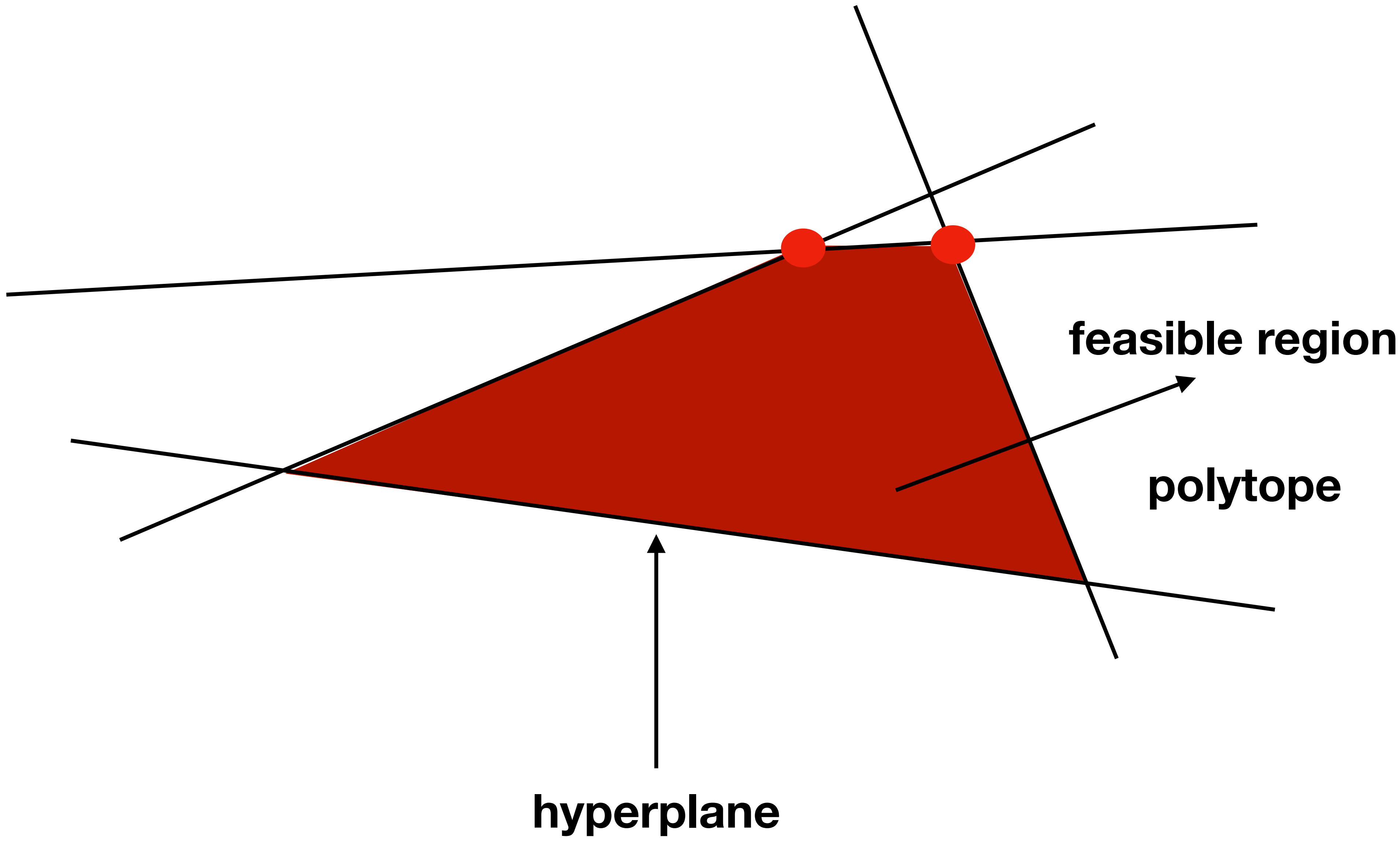
More terminology



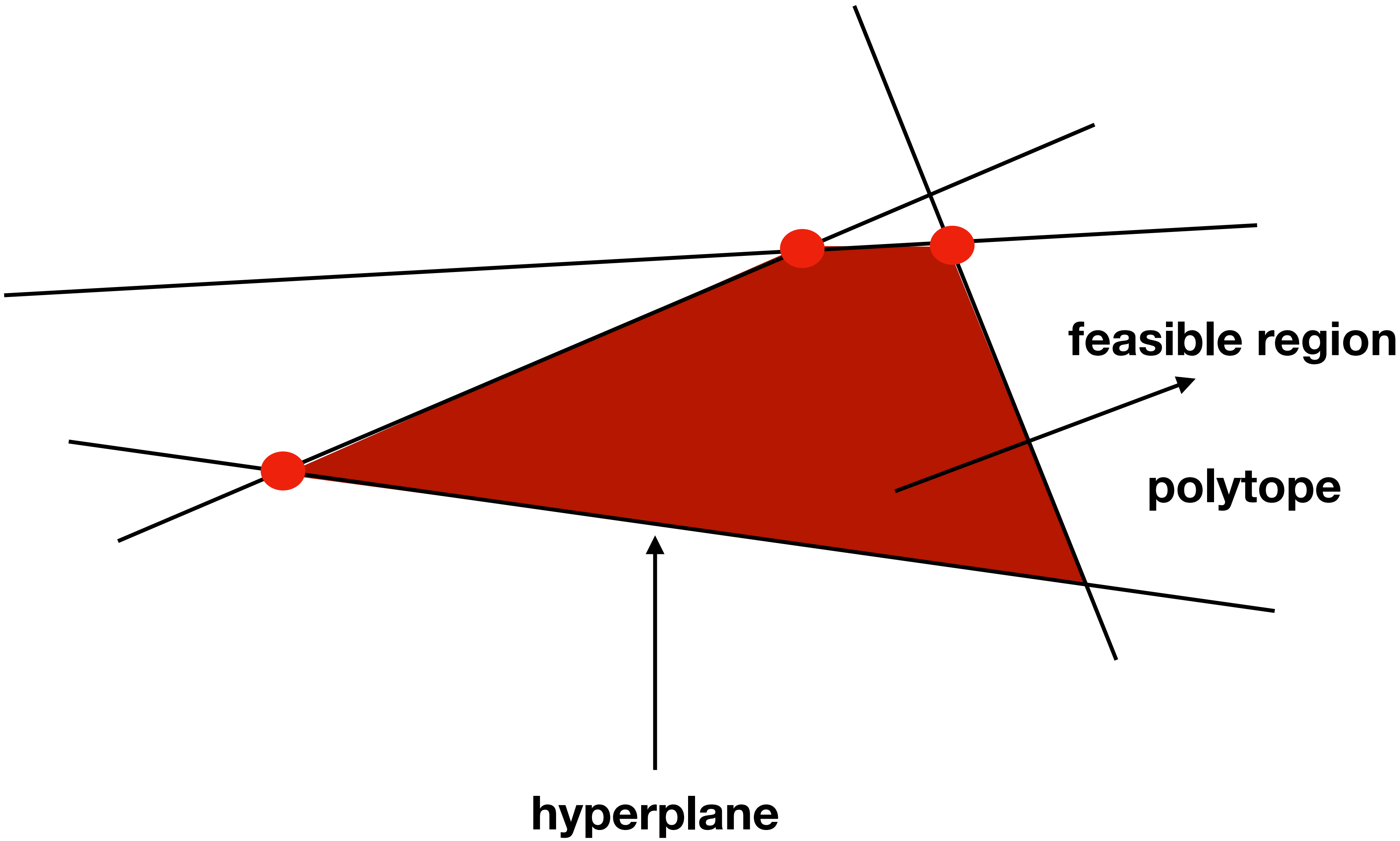
More terminology



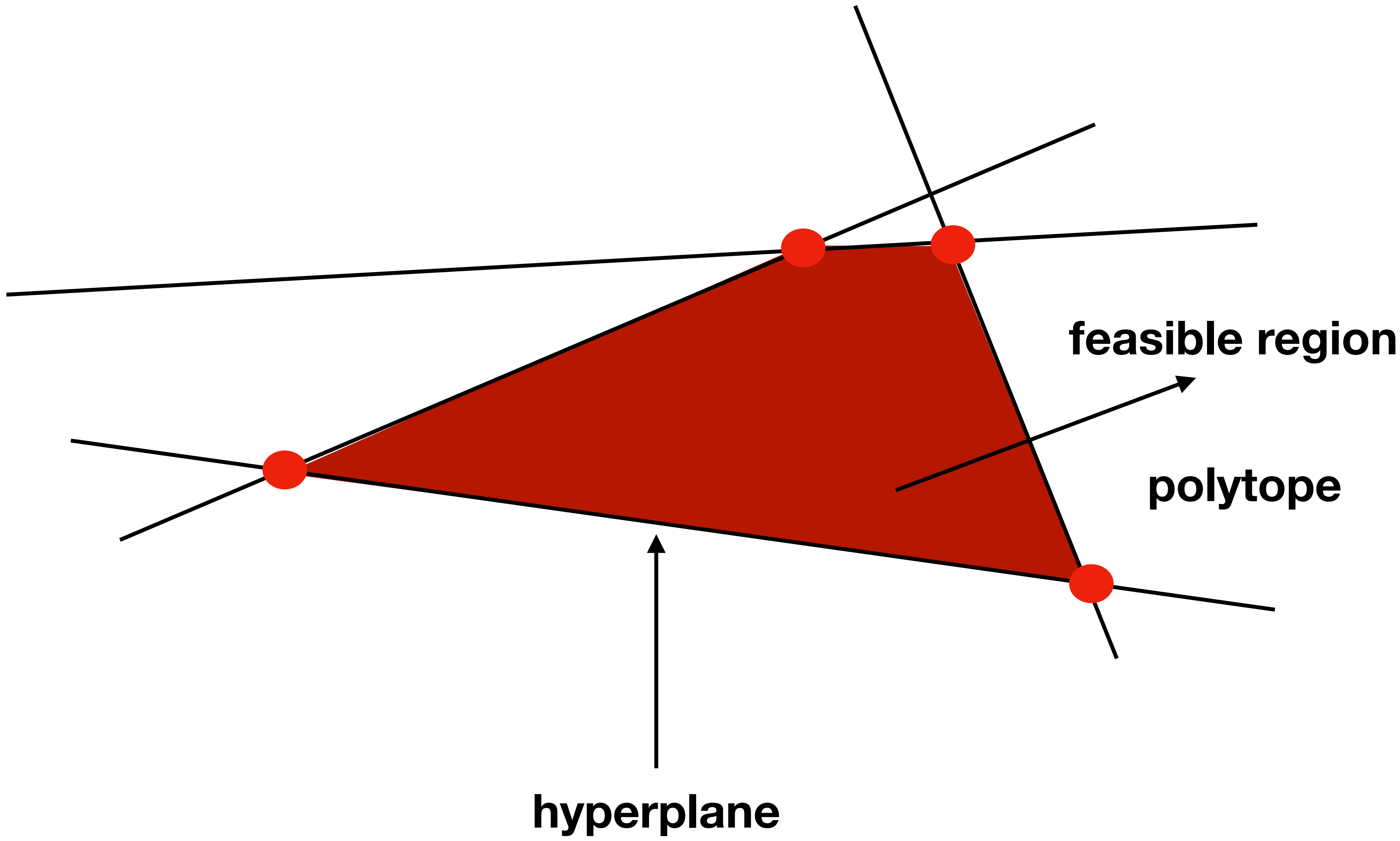
More terminology



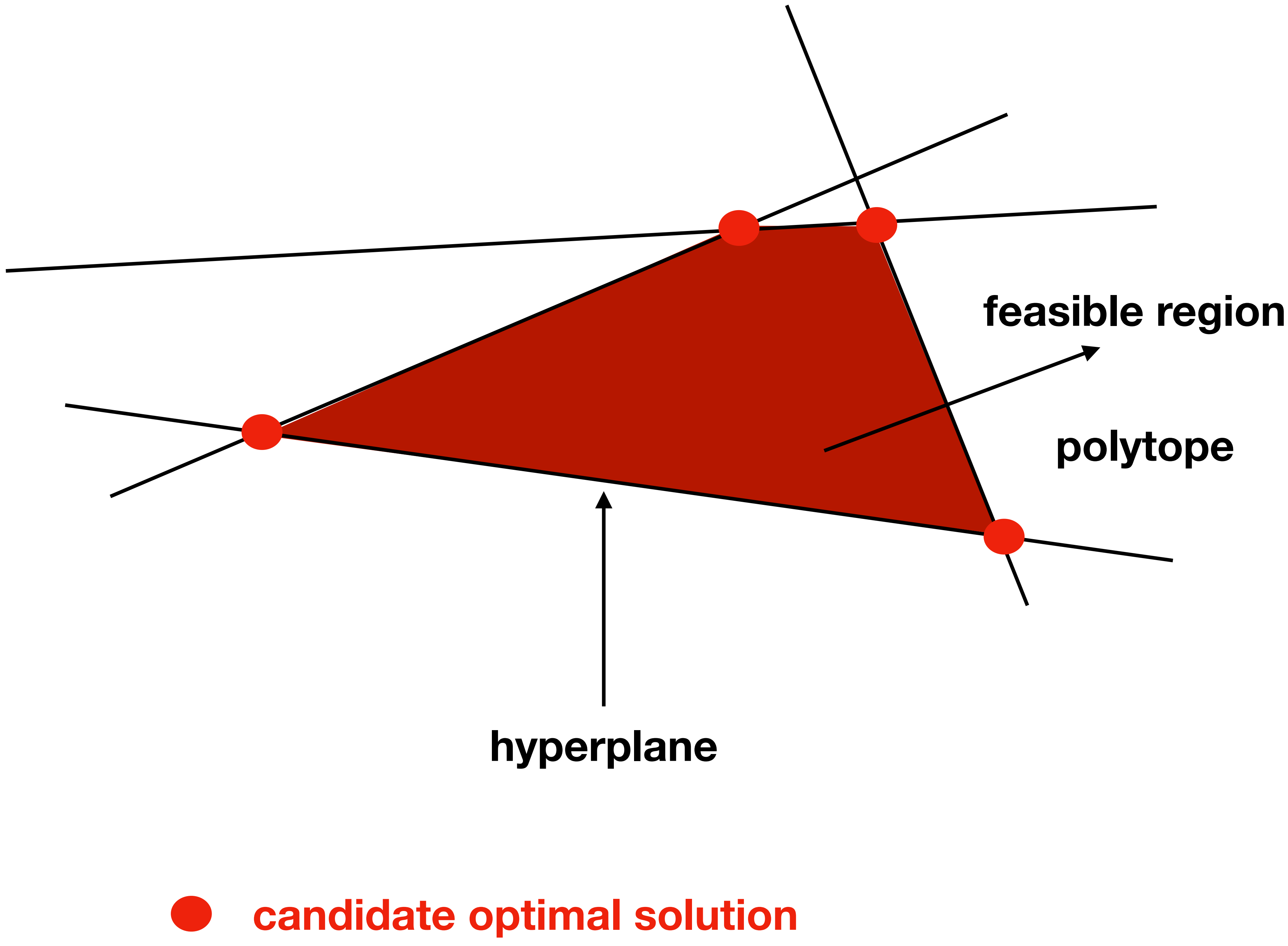
More terminology



More terminology



More terminology



Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

What is the feasible region is empty, or the polytope is not bounded?

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

What is the feasible region is empty, or the polytope is not bounded?

We will consider valid solutions to say that *“the LP is infeasible”* or *“the LP is unbouded”*.

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

This is what the Simplex method does, via *pivoting* (next lecture)

Solving the linear program

To find the optimal solution, it suffices to examine the *corners* of the feasible region.

These are the intersection points of the lines defined by the constraints.

This is what the Simplex method does, via *pivoting* (next lecture)

Other algorithms for solving LPs:
Ellipsoid Method, Interior Point Methods

A simple but inefficient algorithm

A simple but inefficient algorithm

Idea: Elimination of variables

A simple but inefficient algorithm

Idea: Elimination of variables

Observation: Every inequality of the LP can be written in one of two forms

A simple but inefficient algorithm

Idea: Elimination of variables

Observation: Every inequality of the LP can be written in one of two forms

$x_j \geq \alpha$ or $x_j \leq \beta$ for some α, β

A simple but inefficient algorithm

Idea: Elimination of variables

Observation: Every inequality of the LP can be written in one of two forms

$x_j \geq \alpha$ or $x_j \leq \beta$ for some α, β

"Solve" each inequality for x_j.

A simple but inefficient algorithm

Idea: Elimination of variables

Observation: Every inequality of the LP can be written in one of two forms

$x_j \geq \alpha$ or $x_j \leq \beta$ for some α, β

"Solve" each inequality for x_j.

Eliminate x_j from all of the constraints.

A simple but inefficient algorithm

Idea: Elimination of variables

Observation: Every inequality of the LP can be written in one of two forms

$x_j \geq \alpha$ or $x_j \leq \beta$ for some α, β

"Solve" each inequality for x_j.

Eliminate x_j from all of the constraints.

Repeat for the next variable, until we only have one variable.

A simple but inefficient algorithm

Idea: Elimination of variables

Observation: Every inequality of the LP can be written in one of two forms

$x_j \geq \alpha$ or $x_j \leq \beta$ for some α, β

"Solve" each inequality for x_j.

Eliminate x_j from all of the constraints.

Repeat for the next variable, until we only have one variable.

Substitute back to get the other variables.

A simple but inefficient algorithm (example)

A simple but inefficient algorithm (example)

$$
\begin{aligned}
&x + y \geq 0 \\
&2x + y \geq 2 \\
&-x + y \geq 1 \\
&-x + 2y \geq -1
\end{aligned}
$$

A simple but inefficient algorithm (example)

$$x + y \geq 0$$
$$2x + y \geq 2$$
$$-x + y \geq 1$$
$$-x + 2y \geq -1$$

"Solve" for x:

A simple but inefficient algorithm (example)

$$x + y \geq 0$$
$$2x + y \geq 2$$
$$-x + y \geq 1$$
$$-x + 2y \geq -1$$

"Solve" for x:

$$x \geq -y$$
$$x \geq 1 - \frac{y}{2}$$
$$x \leq -1 + y$$
$$x \leq 1 + 2y$$

A simple but inefficient algorithm (example)

$$x \geq -y$$
$$x \geq 1 - \frac{y}{2}$$
$$x \leq -1 + y$$
$$x \leq 1 + 2y$$

A simple but inefficient algorithm (example)

$$x \geq -y$$
$$x \geq 1 - \frac{y}{2}$$
$$x \leq -1 + y$$
$$x \leq 1 + 2y$$

The above implies:

A simple but inefficient algorithm (example)

$$x \geq -y$$
$$x \geq 1 - \frac{y}{2}$$
$$x \leq -1 + y$$
$$x \leq 1 + 2y$$

The above implies:

$$-1 + y \geq -y$$
$$-1 + y \geq 1 - \frac{y}{2}$$
$$1 + 2y \geq -y$$
$$1 + 2y \geq 1 - \frac{y}{2}$$

A simple but inefficient algorithm (example)

$$x \geq -y$$
$$x \geq 1 - \frac{y}{2}$$
$$x \leq -1 + y$$
$$x \leq 1 + 2y$$

Simplifying:

The above implies:

$$-1 + y \geq -y$$
$$-1 + y \geq 1 - \frac{y}{2}$$
$$1 + 2y \geq -y$$
$$1 + 2y \geq 1 - \frac{y}{2}$$

A simple but inefficient algorithm (example)

$$x \geq -y$$
$$x \geq 1 - \frac{y}{2}$$
$$x \leq -1 + y$$
$$x \leq 1 + 2y$$

The above implies:

$$-1 + y \geq -y$$
$$-1 + y \geq 1 - \frac{y}{2}$$
$$1 + 2y \geq -y$$
$$1 + 2y \geq 1 - \frac{y}{2}$$

Simplifying:

$$y \geq 1/2$$
$$y \geq 4/3$$
$$y \geq -1/3$$
$$y \geq 0$$

A simple but inefficient algorithm (example)

Simplifying:

$y \geq 1/2$
$y \geq 4/3$
$y \geq -1/3$
$y \geq 0$

A simple but inefficient algorithm (example)

Pick a feasible y,
e.g., $y = 2$.

Simplifying:

$y \geq 1/2$
$y \geq 4/3$
$y \geq -1/3$
$y \geq 0$

A simple but inefficient algorithm (example)

Simplifying:

$$y \geq 1/2$$
$$y \geq 4/3$$
$$y \geq -1/3$$
$$y \geq 0$$

Pick a feasible y, e.g., $y = 2$.

We can find a feasible x using our inequalities:

$$x \geq -y$$
$$x \geq 1 - \frac{y}{2}$$
$$x \leq -1 + y$$
$$x \leq 1 + 2y$$

A simple but inefficient algorithm (example)

A simple but inefficient algorithm (example)

How do we find an optimal solution?

A simple but inefficient algorithm (example)

How do we find an optimal solution?

Observation: Given a linear objective function, we can substitute it with a variable x_0 (how?)

Diet Example

Minimise $12x + 15y$

subject to

$$x + y \geq 5$$
$$2x + y \geq 6$$
$$x + 3y \geq 9$$
$$x, y \geq 0$$

Diet Example

Minimise x_0

subject to

$$x + y \geq 5$$
$$2x + y \geq 6$$
$$x + 3y \geq 9$$
$$x, y \geq 0$$
$$12x + 15y = x_0$$

A simple but inefficient algorithm (example)

How do we find an optimal solution?

Observation: Given a linear objective function, we can substitute it with a variable x_0 (how?)

A simple but inefficient algorithm (example)

How do we find an optimal solution?

Observation: Given a linear objective function, we can substitute it with a variable x_0 (how?)

Eliminate to find inequalities for x_0.

A simple but inefficient algorithm (example)

How do we find an optimal solution?

Observation: Given a linear objective function, we can substitute it with a variable x_0 (how?)

Eliminate to find inequalities for x_0.

Pick the x_0 that optimises the objective function.

A simple but inefficient algorithm (example)

How do we find an optimal solution?

Observation: Given a linear objective function, we can substitute it with a variable x_0 (how?)

Eliminate to find inequalities for x_0.

Pick the x_0 that optimises the objective function.

Work out feasible $x_1, \ldots, x_n$ for the rest of the variables.

A simple but inefficient algorithm (example)

A simple but inefficient algorithm (example)

The algorithm is called Fourier-Motzkin Elimination (1826, 1936).

A simple but inefficient algorithm (example)

The algorithm is called Fourier-Motzkin Elimination (1826, 1936).

Similar idea to Gaussian Elimination.

A simple but inefficient algorithm (example)

The algorithm is called Fourier-Motzkin Elimination (1826, 1936).

Similar idea to Gaussian Elimination.

Simple but highly inefficient: One elimination step over m inequalities can result in $\Omega(n^2)$ new inequalities.

A simple but inefficient algorithm (example)

The algorithm is called Fourier-Motzkin Elimination (1826, 1936).

Similar idea to Gaussian Elimination.

Simple but highly inefficient: One elimination step over m inequalities can result in $\Omega(n^2)$ new inequalities.

Thus for k elimination steps we can have $\Omega\left(m^{2^k}\right)$ constraints.

A nice consequence of FME

If the LP has an optimal feasible solution, then it has a rational optimal feasible solution x^* and the objective function value $f(x^*)$ is also rational.

Linear programming (LP)

$$\text{maximise} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1\in}^{n} \alpha_{ij} x_j \leq b_i, \quad i = 1, ..., m$$

$$x_j \geq 0, \quad j = 1, ..., n$$

Integer Linear programming

$$\text{maximise} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1 \in}^{n} \alpha_{ij} x_j \leq b_i, \quad i = 1, ..., m$$

$$x_j \geq 0, \quad j = 1, ..., n$$

$$x_j \text{ is integer}$$

Integer Linear programming

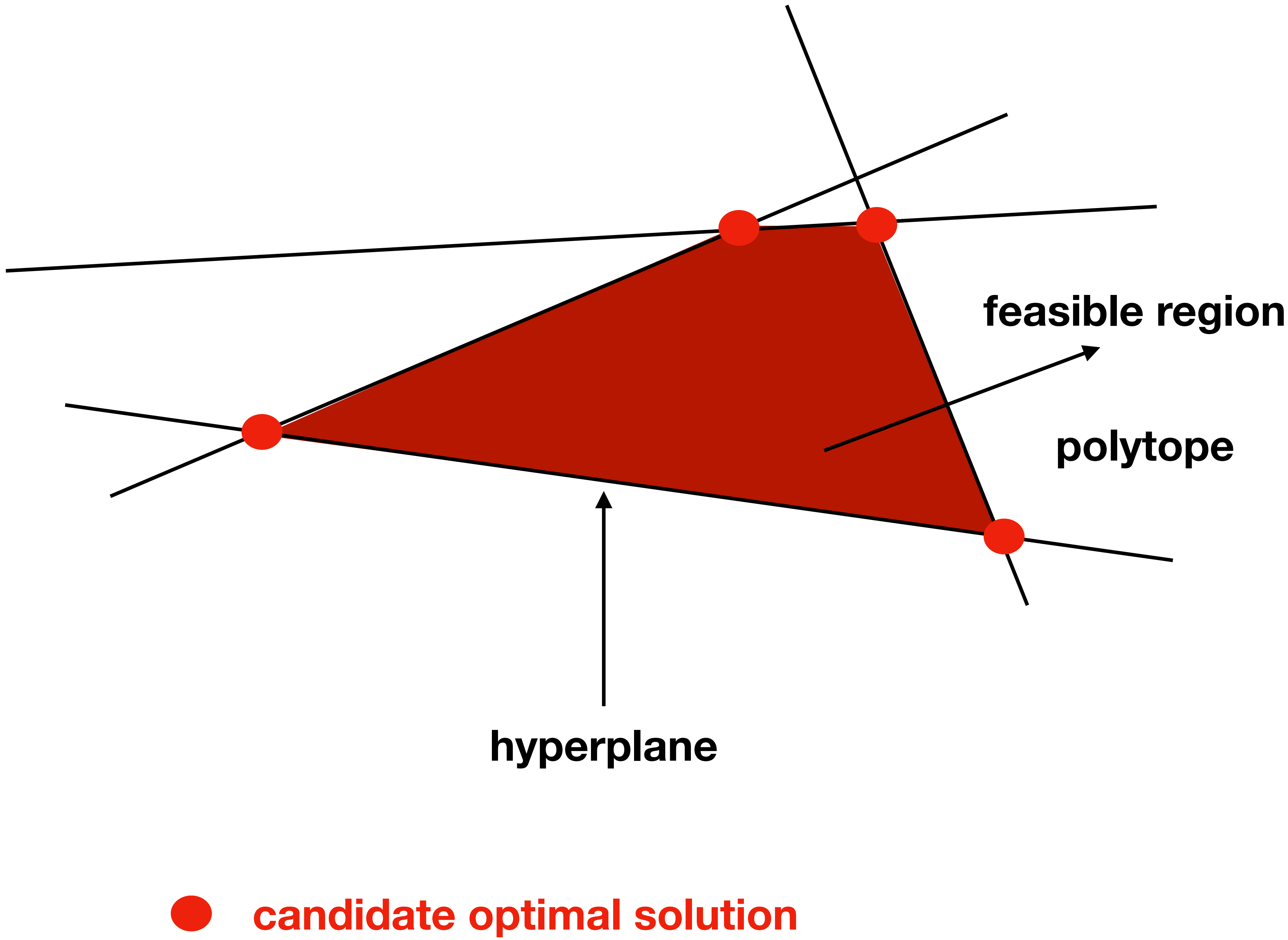
$$\text{maximise} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1\in}^{n} \alpha_{ij} x_j \leq b_i, \quad i = 1, ..., m$$

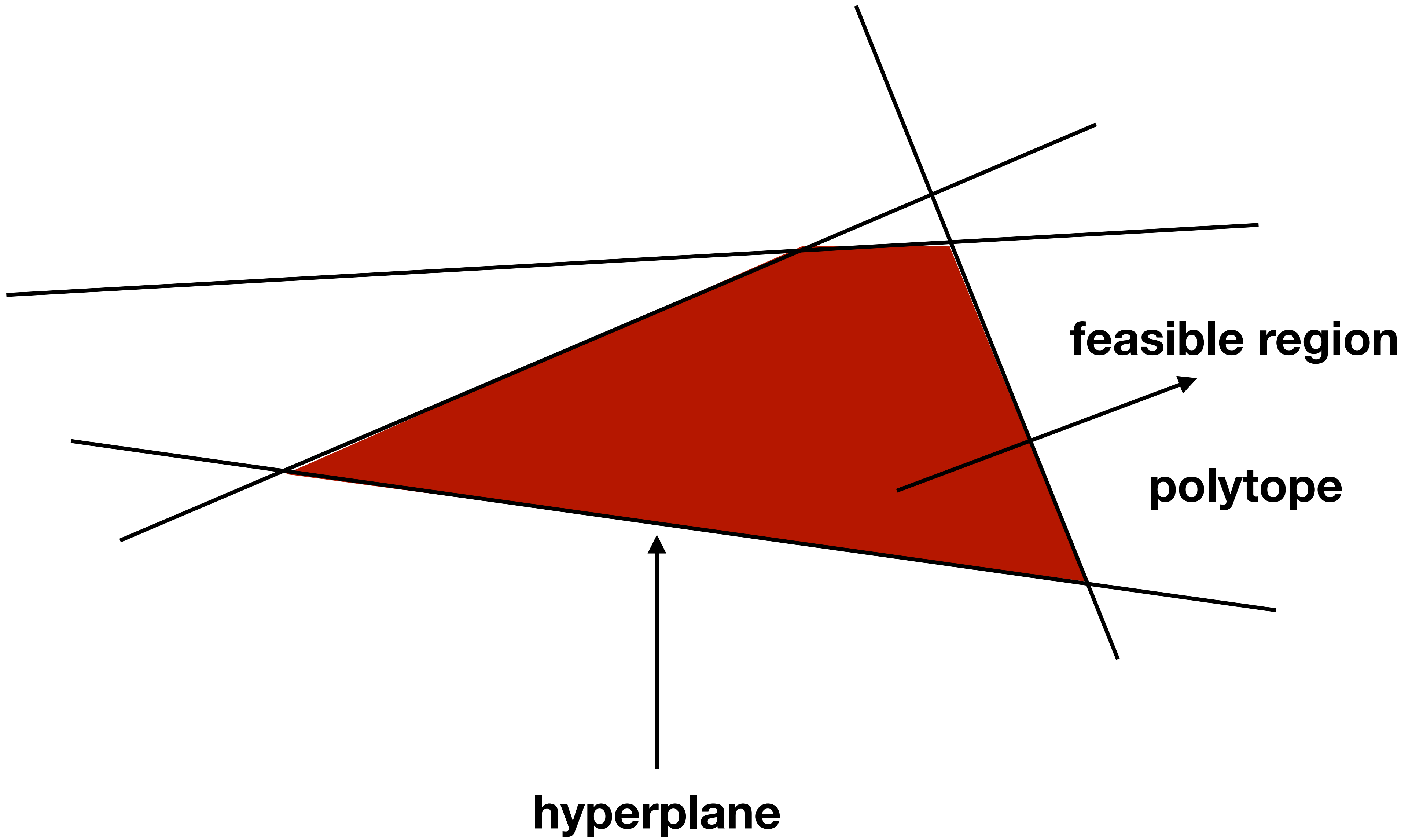
$$x_j \geq 0, \quad j = 1, ..., n$$

$$\boxed{x_j \text{ is integer}}$$

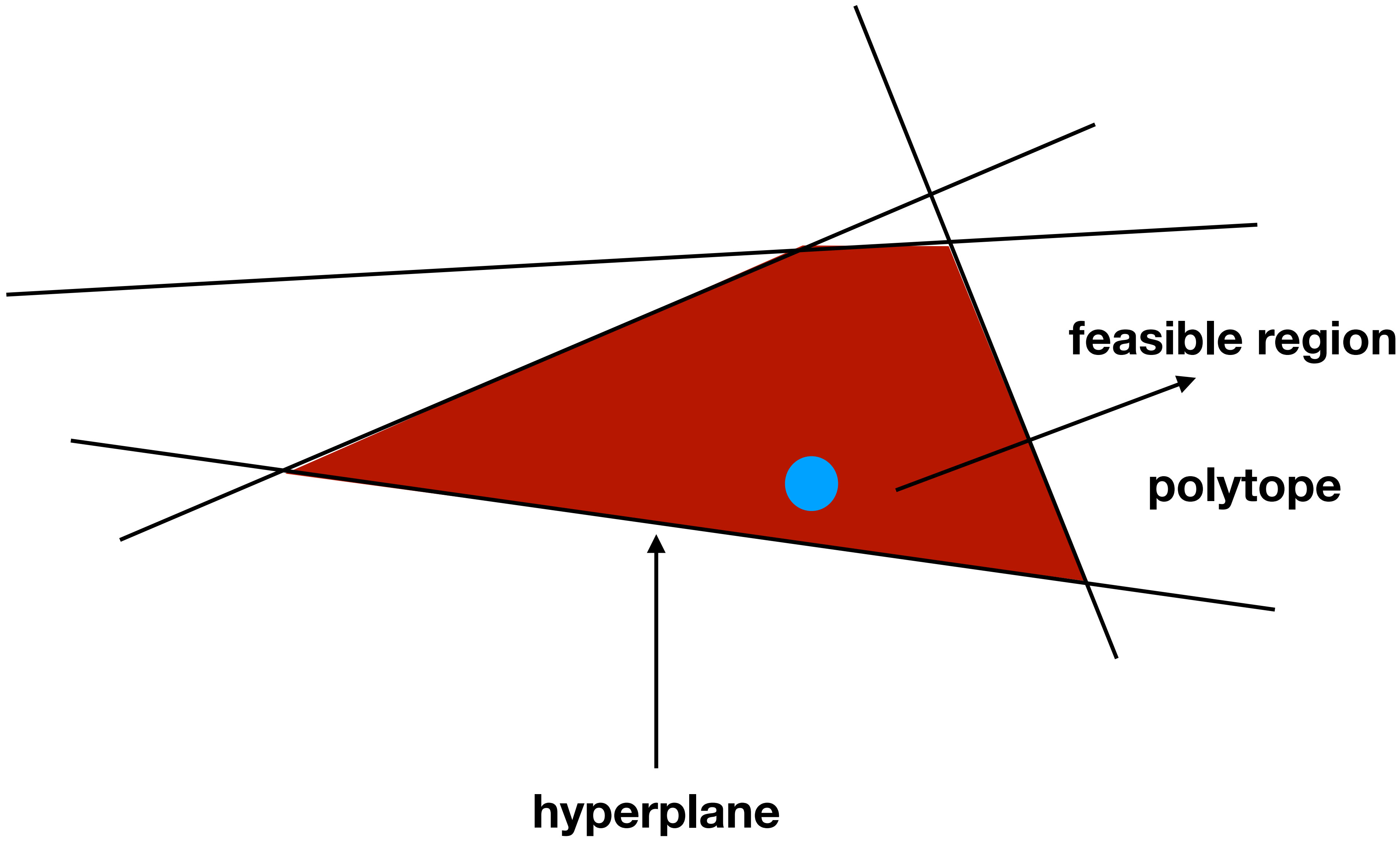
Feasible region



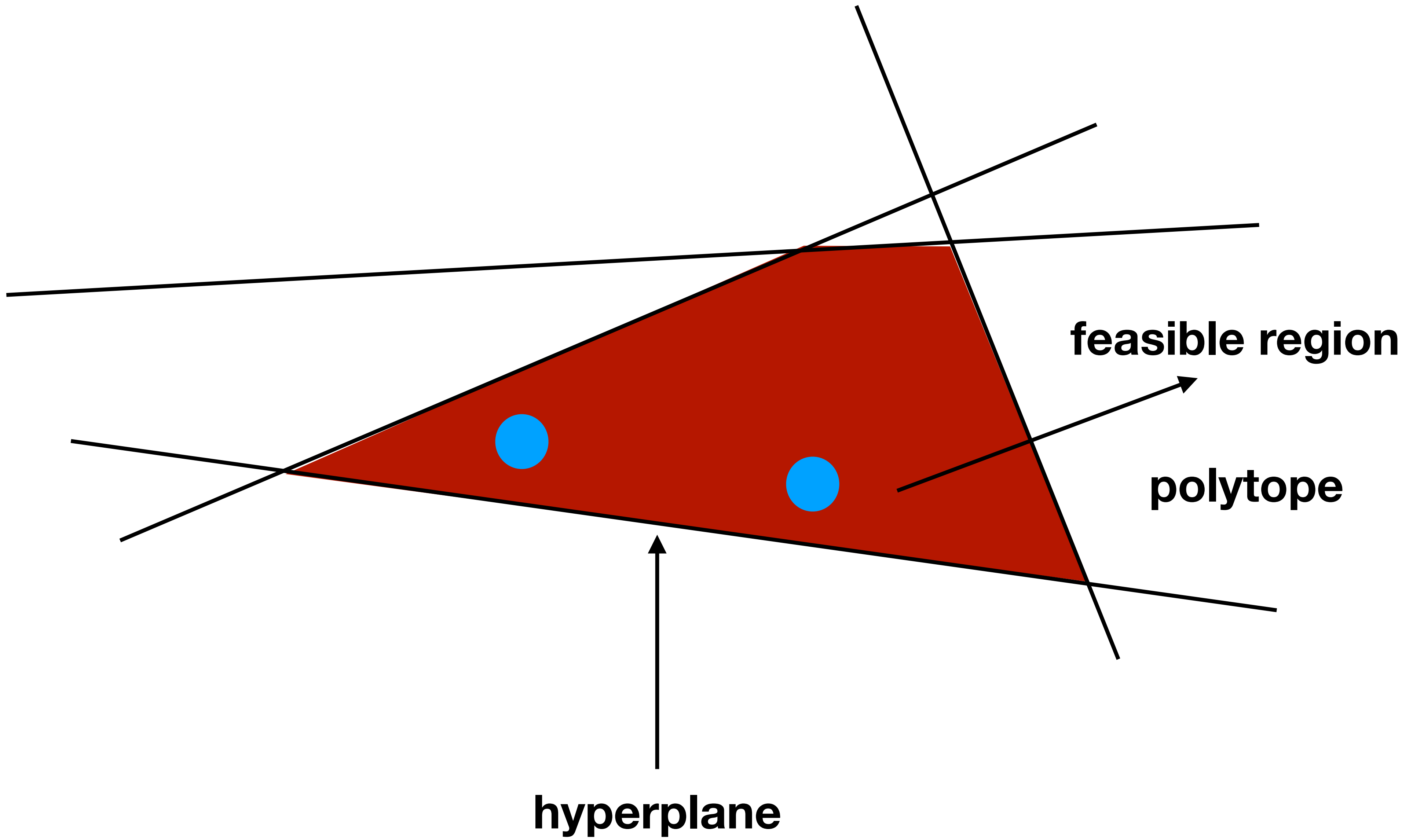
Feasible region



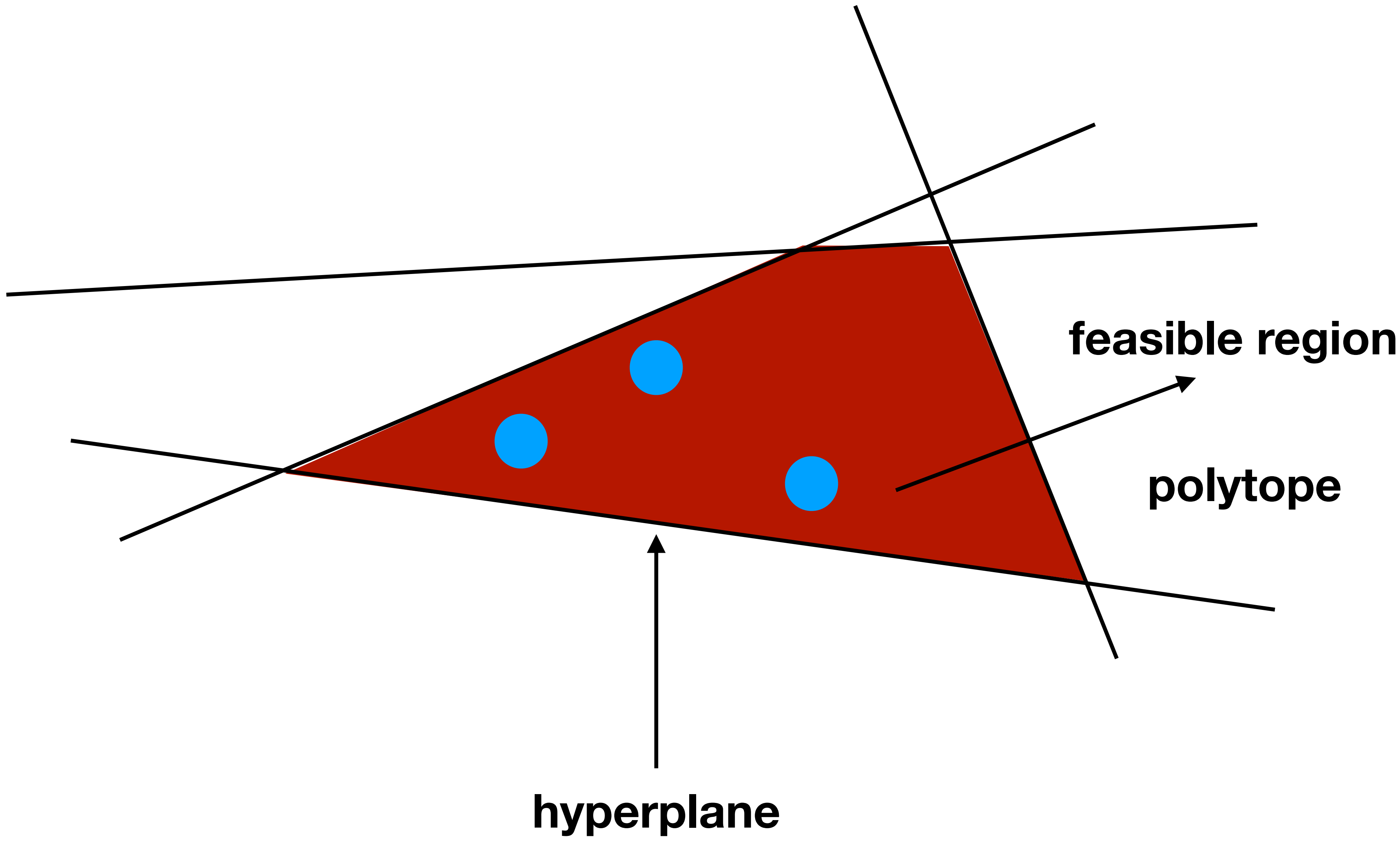
Feasible region



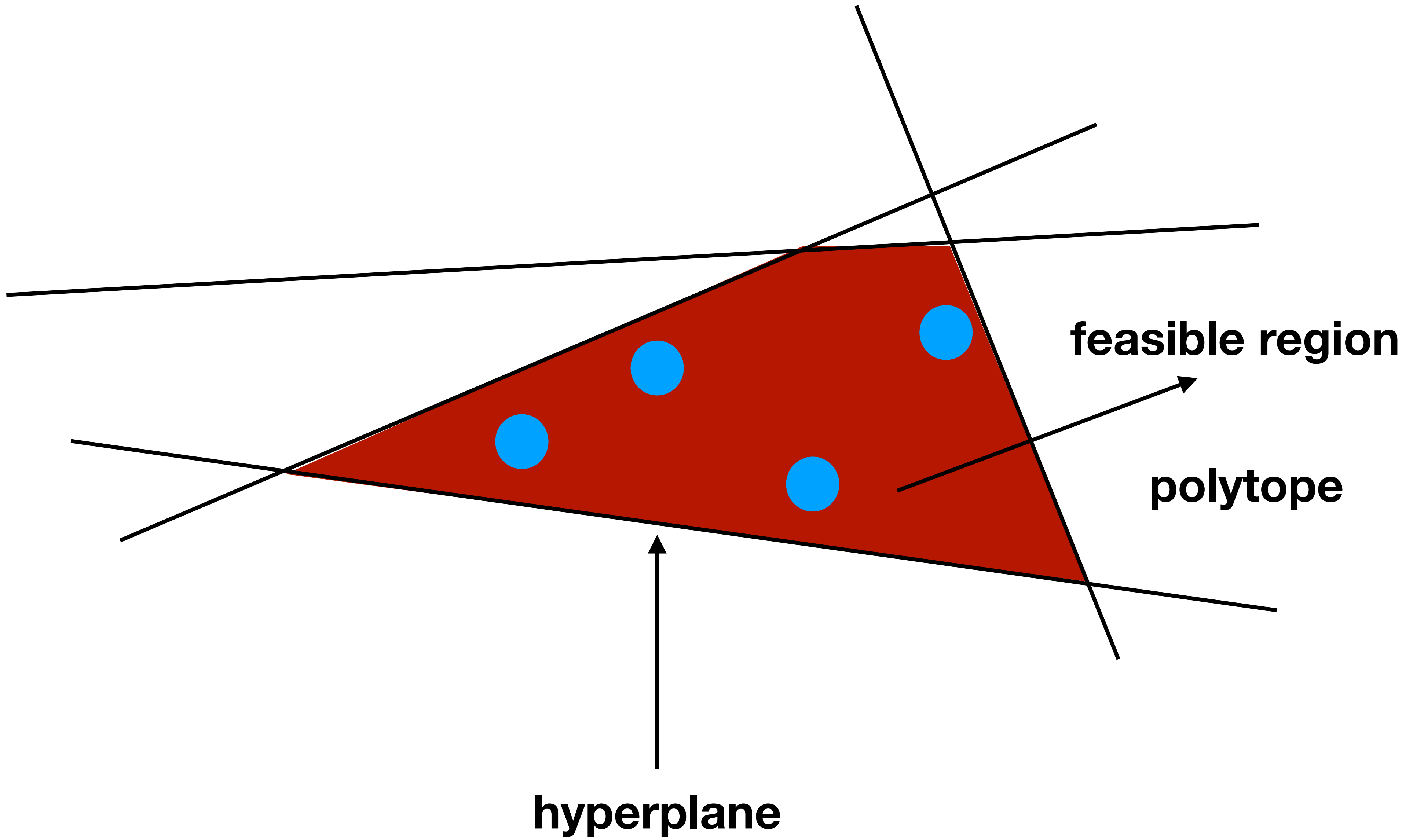
Feasible region



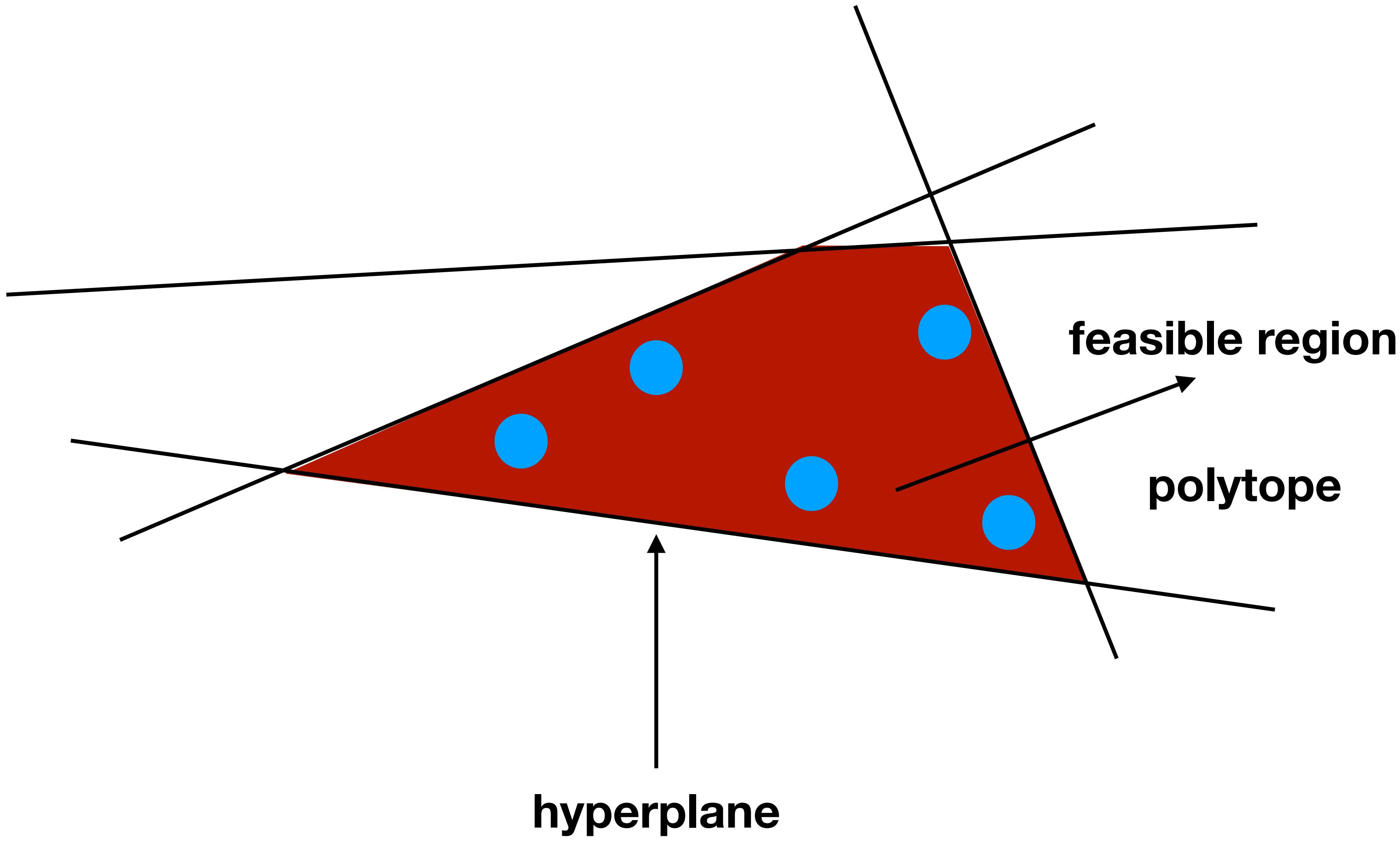
Feasible region



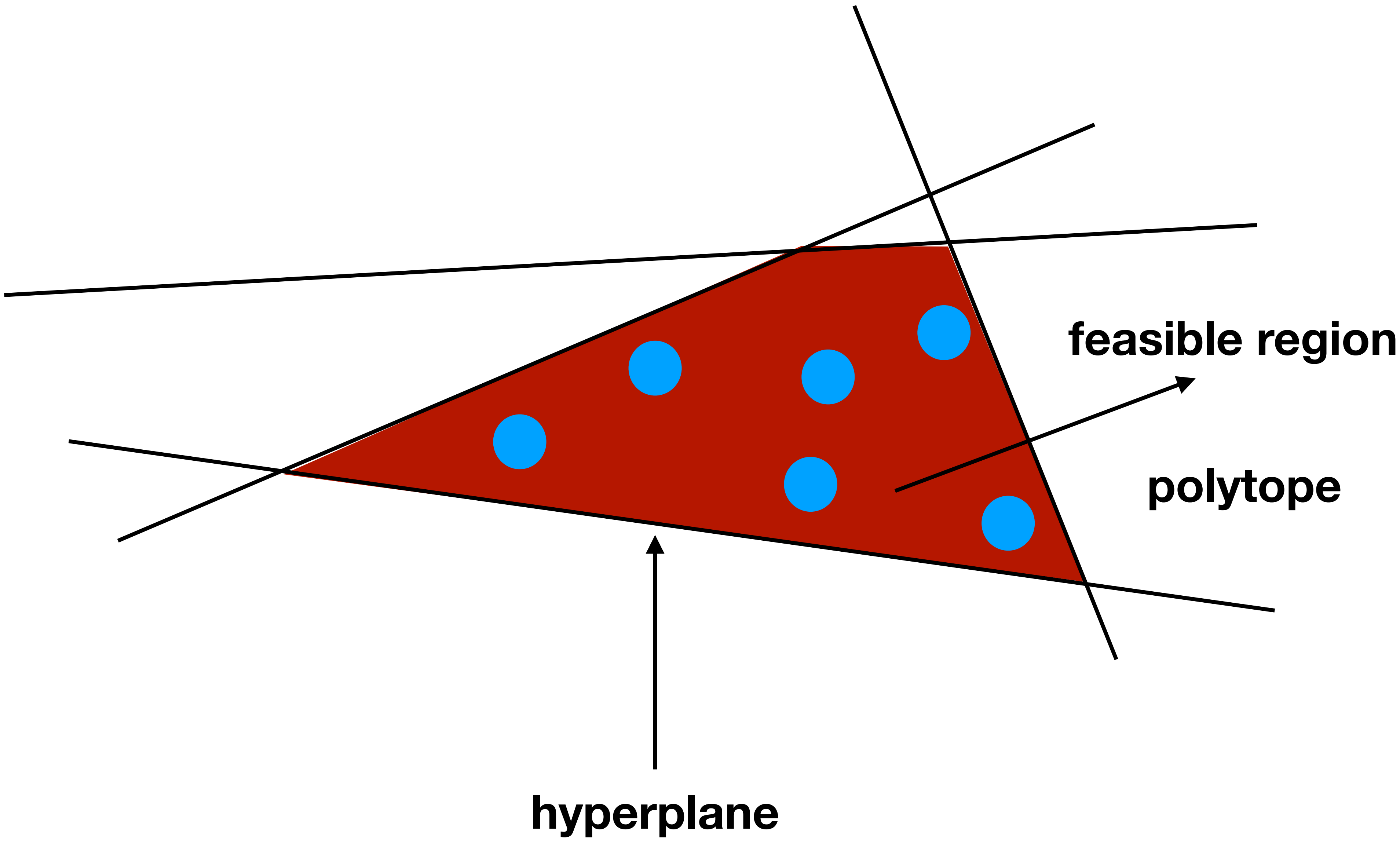
Feasible region



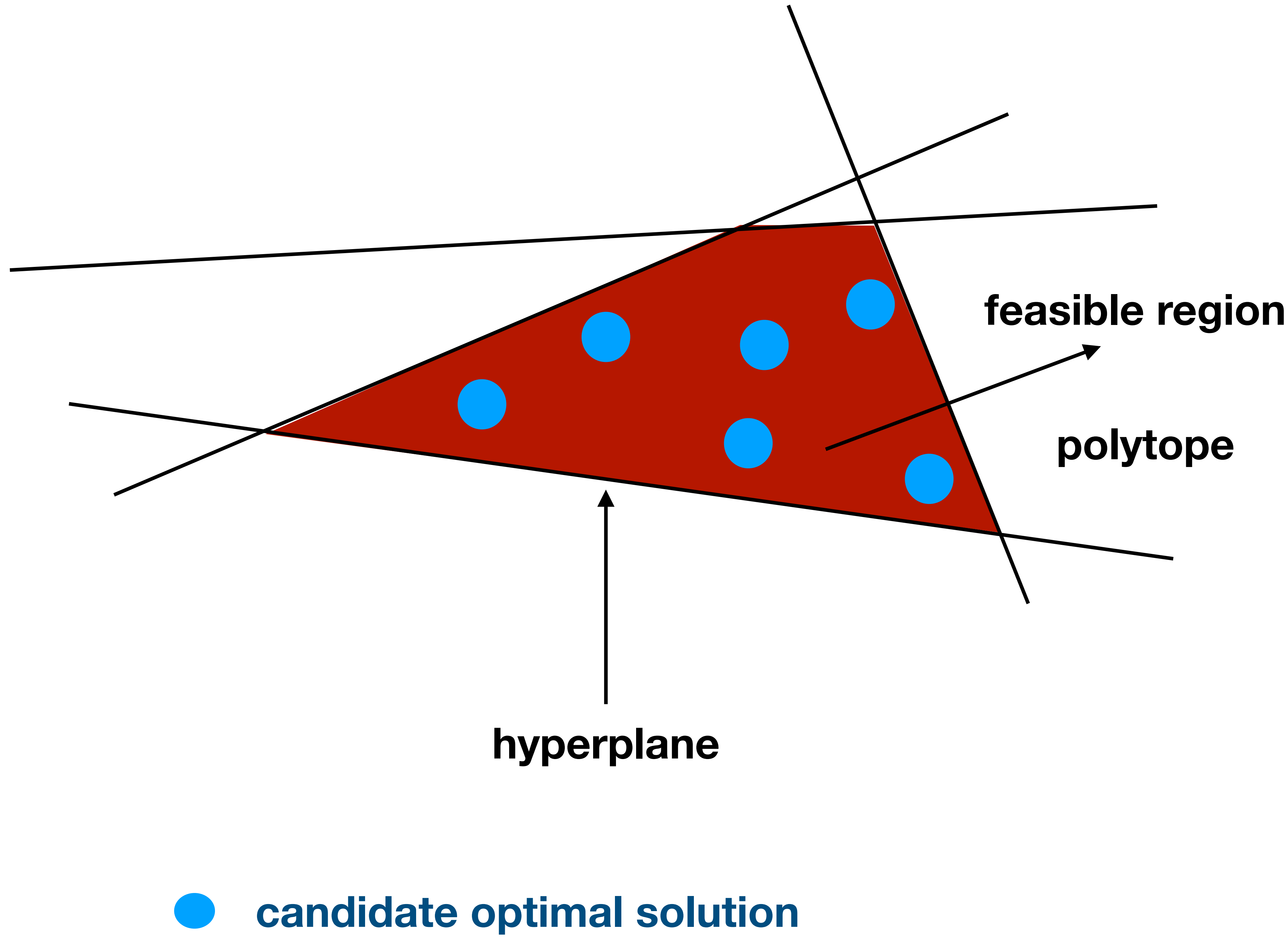
Feasible region



Feasible region



Feasible region



Solving ILPs

Solving ILPs

The corners are not necessarily integer solutions.

Solving ILPs

The corners are not necessarily integer solutions.

It does not suffice to look at the corners.

Solving ILPs

The corners are not necessarily integer solutions.

It does not suffice to look at the corners.

We can exhaustively try all possible integer solutions.

Solving ILPs

The corners are not necessarily integer solutions.

It does not suffice to look at the corners.

We can exhaustively try all possible integer solutions.

Can we do something more clever?

Solving ILPs

The corners are not necessarily integer solutions.

It does not suffice to look at the corners.

We can exhaustively try all possible integer solutions.

Can we do something more clever?

Yes, but in the worst-case, it will still take exponential time in many ILPs.

Solving ILPs

The corners are not necessarily integer solutions.

It does not suffice to look at the corners.

We can exhaustively try all possible integer solutions.

Can we do something more clever?

Yes, but in the worst-case, it will still take exponential time in many ILPs.

Generally speaking, ILP solving is NP-hard.

Summarising

Summarising

Linear Programs can be solved in polynomial time.

Summarising

Linear Programs can be solved in polynomial time.

Ellipsoid method, interior point methods.

Summarising

Linear Programs can be solved in polynomial time.

Ellipsoid method, interior point methods.

We will not learn how these work, this is for a course on optimisation.

Summarising

Linear Programs can be solved in polynomial time.

Ellipsoid method, interior point methods.

We will not learn how these work, this is for a course on optimisation.

Not the Simplex Algorithm!

Summarising

Linear Programs can be solved in polynomial time.

Ellipsoid method, interior point methods.

We will not learn how these work, this is for a course on optimisation.

Not the Simplex Algorithm!

But we will learn about this algorithm in the next lecture, because of its very important principles.

Summarising

Linear Programs can be solved in polynomial time.

Ellipsoid method, interior point methods.

We will not learn how these work, this is for a course on optimisation.

Not the Simplex Algorithm!

But we will learn about this algorithm in the next lecture, because of its very important principles.

Integer Linear Programs generally cannot be solved in polynomial time (unless P=NP).