MATH 652: Optimization II

Part 16

Linear programming 1

$$\text{minimize} \quad c^T x$$
$$Ax \geq b$$

Linear Programming

Definition: A linear program is an optimization problem in which:

- The objective function is linear (affine):

$$f(x) = c^T x + d, \qquad c, x \in \mathbb{R}^n$$

(Note: The constant *d* does not affect where the optimum lies.)

- All equality and inequality constraints are linear (affine):

$$a_i^T x \geq b_i, \qquad a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$$

or

$$a_i^T x = b_i, \qquad a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$$

Note: An equality constraint is equivalent to two inequality constraints:

$$a_i^T x \geq b_i,$$
$$a_i^T x \leq b_i.$$

Linear Programming

Definition: A linear program is an optimization problem that can be written equivalently as

$$\min_{x \in \mathbb{R}^n} \quad c^T x$$
$$Ax \geq b$$

Linear Programming: Example 1

Production planning:

A company has products $i=1...N$. Sales price: c_i.

Product i needs a_{ji} units of resource $j=1...M$. Availability: b_j units.

Question: How much of each product should the company produce?

Mathematical formulation:

- x_i: Items of product i to be produced
- Revenue: $\sum_{i=1}^{P} c_i x_i = c^T x$
- We need $\sum_{i=1}^{P} a_{ji} x_i = a_j^T x$ units of resource j
- Optimization problem:

$$\boxed{\begin{aligned} &\max_{x \in \mathbb{R}^n} \quad c^T x \\ &\qquad\quad a_j^T x \leq b_j, \qquad j=1\ldots M \end{aligned}}$$

or equivalently

$$\boxed{\begin{aligned} &\min_{x \in \mathbb{R}^n} \quad -c^T x \\ &\qquad\quad -A x \geq -b \end{aligned}}$$

Linear Programming: Example 2

Future capacity planning: A power company foresees a demand of d_t gigawatt-hours in year t=*2019..2075*. Existing capacity: e_t.

Choice: Extend capacity using coal (price: c_t per gigawatt-hour when built in year t) or wind (cost: w_t). Minimize costs.

Constraints: (i) Coal power plants last for 25 years, wind plants for 30. (ii) Meet foreseen demand.

Mathematical formulation:

- x_t,y_t: capacity of coal and wind power built in year t
- Total cost:

$$\sum_{t=2010}^{2075} c_t x_t + w_t y_t$$

- Available capacity in year t:

$$e_t + \sum_{s=\max\{2019,t-25\}}^{t} x_s + \sum_{s=\max\{2019,t-30\}}^{t} y_s$$

Linear Programming: Example 2

Mathematical formulation:

$$
\begin{array}{lll}
\min_{x_t,\, y_t} & \sum_{t=2019}^{2075} c_t x_t + w_t y_t & \\
 & \sum_{s=\max\{2019,t-25\}}^{t} x_s + \sum_{s=\max\{2019,t-30\}}^{t} y_s \geq d_t - e_t, & t=2019..2075 \\
 & x_t \geq 0, & t=2019..2075 \\
 & y_t \geq 0, & t=2019..2075
\end{array}
$$

Reformulate in standard form by using

$$X=(x_{2019}, x_{2020}, \ldots, x_{2075}, \quad y_{2019}, y_{2020}, \ldots, y_{2075})^T$$

$$C=(c_{2019}, c_{2020}, \ldots, c_{2075}, \quad w_{2019}, w_{2020}, \ldots, w_{2075})^T$$

$$b=(d_{2019}\text{-}e_{2019},\ d_{2020}\text{-}e_{2020},\ \ldots,\ d_{2075}\text{-}e_{2075})^T$$

$$
\begin{array}{ll}
\min_{x \in \mathbb{R}^{82}} & C^T X \\
 & A\,X \geq b \\
 & X \geq 0
\end{array}
$$

Linear Programming: Example 3

Scheduling of resources to tasks: Hospital needs to schedule nurses to night shifts. Nurses work 5 days in a row on a 7-day schedule. On the *i*th day of the week, history shows that d_i nurses are needed (demand).

Question: How many nurses are needed in total, and on what schedules should they be?

Mathematical formulation:

- x_i: number of nurses starting their 5-day run on day i
- Total number of nurses needed:

$$\sum_{i=1}^{7} x_i$$

- On day 1, the number of nurses available is

$$x_4 + x_5 + x_6 + x_7 + x_1$$

- On day 2, the number of nurses available is

$$x_5 + x_6 + x_7 + x_1 + x_2$$

Linear Programming: Example 3

Mathematical formulation:

$$\min_{x \in \mathbb{R}^7} \sum_{i=1}^{7} x_i$$

$$\sum_{k=i-4}^{i} x_{i \bmod 7} \geq d_i, \qquad i = 1 \ldots 7$$

$$x_i \geq 0, \qquad i = 1 \ldots 7$$

Note: More realistic formulations would also include preferences by employees, conflicts, tied schedules (e.g. advisor/trainee), contingencies/on-call duties, etc.

Linear Programming: Example 3

Mathematical formulation:

$$
\begin{array}{llll}
\min_{x \in \mathbb{R}^7} & \sum_{i=1}^{7} x_i & & \\
& \sum_{k=i-4}^{i} x_{i \bmod 7} & \geq d_i, & i=1\ldots7 \\
& x_i & \geq 0, & i=1\ldots7
\end{array}
$$

However: In reality, we also need to consider that the number of nurses, x_i, must be an integer.

The problem is therefore not a common *linear programming* (LP) problem, but an *integer linear programming* (ILP) problem.

ILPs are much more difficult problems to solve!

Linear Programming: Example 4

Network flow: Food needs to get from a set of locations ("sources") of capacity s_i to a different set of locations ("sinks") of demand d_i. Sources have d_i=0, sinks s_i=0. Transportation happens on a network between nodes *(i,j)* of bandwidth b_{ij} and transporting one unit of food on this link costs c_{ij}.

How to transport food most cost effective so that demand is met?

Linear Programming: Example 4

Mathematical formulation:

- x_{ij}: amount of food transported on edge *(i,j)* of the network, i.e. from *i* to *j*

- Need to satisfy bandwidth constraints for each link:

$$x_{ij} \le b_{ij}$$

(Note: If a link does not exist, then $b_{ij}=0$. Links are *directed*!)

- Sources can not deliver more than they have:

$$\sum_j x_{ij} \le s_i \qquad \text{at sources}$$

- Sinks need to get their demand satisfied:

$$\sum_i x_{ij} \ge d_j \qquad \text{at sinks}$$

Linear Programming: Example 4

Mathematical formulation:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^{n\times n}} \quad & \sum_{i,j} c_{ij} x_{ij} \\
x_{ij} &\leq b_{ij}, \\
x_{ij} &\geq 0, \\
\sum_j x_{ij} &\leq s_i, \\
\sum_i x_{ij} &\geq d_j .
\end{aligned}
$$

Note: Such problems appear in a wide variety of transportation problems. Examples: Shipping of books from amazon.com's distribution centers to customers; supplying goods from King Sooper's distribution centers to stores; etc.

A more complex version of the network flow problem would include a variety of products, rather than just one, or that origin and destination of each product matter (e.g. letters).

Linear Programming: Example 5

Network capacity: Data packets need to get from node A to node B along a network with given bandwidth b_{ij} on each edge.

What is the maximal data rate that can be transported on this network from A to B?

National LambdaRail network, part of NSF's TeraGrid.

Linear Programming: Example 5

Mathematical formulation:

- x_{ij}: data rate transported on edge *(i,j)* of the network, i.e. from *i* to *j*

- Bandwidth constraints for each link:

$$x_{ij} \leq b_{ij}$$

(Note: If a link does not exist, then $b_{ij}=0$. Links are *directed*!)

- At $i=A$, we have for the net inbound flux:

$$\sum_k x_{kA} - \sum_j x_{Aj} = -s$$

- At $i=B$, we have:

$$\sum_k x_{kB} - \sum_j x_{Bj} = s$$

- At all other nodes, data is just transported through:

$$\sum_k x_{ki} - \sum_j x_{ij} = 0$$

Linear Programming: Example 5

Mathematical formulation:

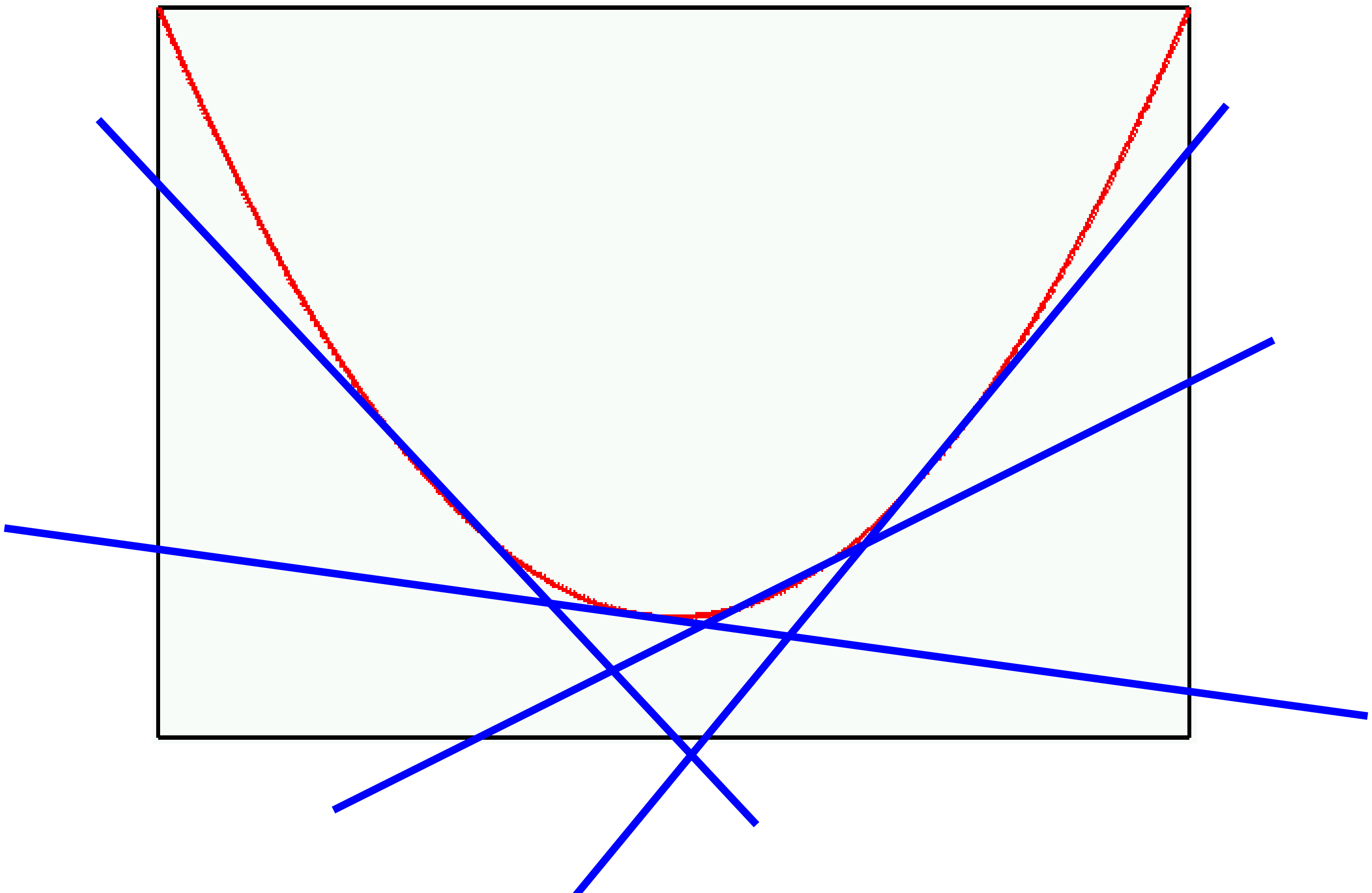
$$
\begin{array}{lll}
\min_{x\in\mathbb{R}^{n\times n},\, s\in\mathbb{R}} & s & \\
 & x_{ij} & \le\ b_{ij}, \\
 & x_{ij} & \ge\ 0, \\
 & \sum_k x_{ki} - \sum_j x_{ij} & =\ s\left(-\delta_{iA} + \delta_{iB}\right)
\end{array}
$$

Note: Extensions might consider that there can be multiple sources but only one destination (e.g. Google data centers, Netflix servers), or that multiple products need to be provided.

Network capacity problems are ubiquitous in transportation planning such as highway networks, airplane scheduling, etc.

Linear Programming: Example 6

Convex optimization of a nonlinear function: Consider finding the minimum of a function *f(x)* that may be nonlinear. If it is at least convex, we may be able to approximate it with piecewise linears.



Linear Programming: Example 6

Mathematical formulation:

- Original problem:

$$\min_{x \in \mathbb{R}^n} \; f(x)$$

- Equivalent formulation of this is:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n, z \in \mathbb{R}} \quad z \\ &\qquad\qquad\qquad z \;\geq\; f(x) \end{aligned}$$

- An approximation to this can be found if we choose points ξ_i and solve instead

$$\boxed{\begin{aligned} &\min_{x \in \mathbb{R}^n, z \in \mathbb{R}} \quad z \\ &\qquad\qquad\qquad z \;\geq\; f(\xi_i) + \nabla f(\xi_i)^T (x - \xi_i), \qquad \forall i \end{aligned}}$$

Note: The solution of this problem could serve as a good starting point for the full nonlinear minimization. Constraints can easily be incorporated if they are linear, or a linearized as well.

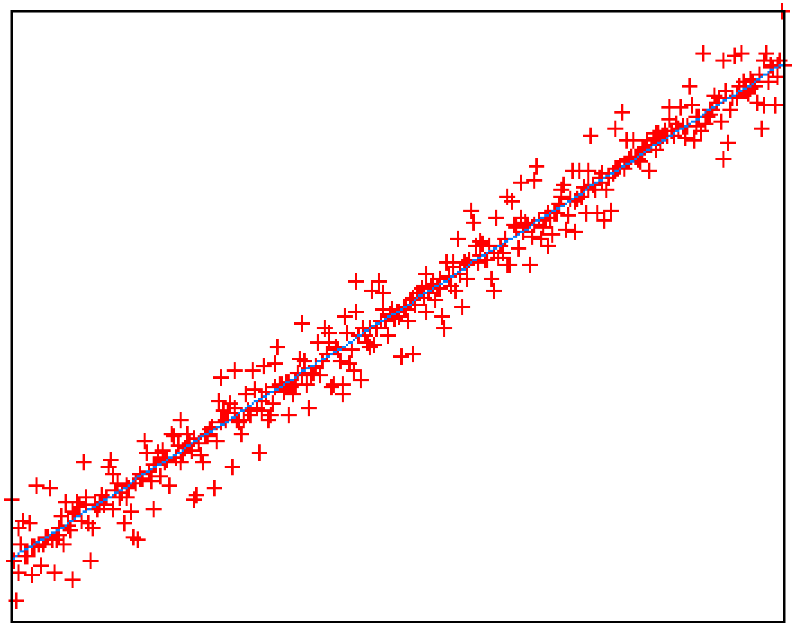
Linear Programming: Example 7

Data fitting: Find a linear relationship that best fits a set of data points. This can be formulated in a variety of ways:

$$\min_{a,b\in\mathbb{R}} \ \frac{1}{2}\sum_i (y_i-(at_i+b))^2$$

$$\min_{a,b\in\mathbb{R}} \ \sum_i |y_i-(at_i+b)|$$

$$\min_{a,b\in\mathbb{R}} \ \max_i |y_i-(at_i+b)|$$



The first one is a smooth, convex problem.The other two are non-smooth but convex problems that can be reformulated as linear programming problems.

Linear Programming: Example 7

Mathematical formulation:

- Original problem:

$$\min_{a,\,b\in\mathbb{R}} \sum_{i=1}^{N} \left| y_i - (a t_i + b) \right|$$

- Equivalent but still non-smooth formulation:

$$\begin{aligned} &\min_{a,\,b\in\mathbb{R},\, s\in\mathbb{R}^n} \sum_{i=1}^{N} s_i \\ &s_i = \left| y_i - (a t_i + b) \right|, \qquad i = 1 \ldots N \end{aligned}$$

- Equivalent but smooth formulation:

$$\boxed{\begin{aligned} &\min_{a,\,b\in\mathbb{R},\, s\in\mathbb{R}^n} \sum_{i=1}^{N} s_i \\ &s_i \geq \left(y_i - (a t_i + b) \right), \qquad i = 1 \ldots N \\ &s_i \geq -\left(y_i - (a t_i + b) \right), \qquad i = 1 \ldots N \end{aligned}}$$

Note: A similar techniques also works for the maximum-residual problem.

Formulating linear programs

Theorem:
Any linear optimization problem that is given in the form

$$\min_{x\in\mathbb{R}^n} \quad c^T x$$
$$A_1 x \geq b_1$$
$$A_2 x = b_2$$
$$A_3 x \leq b_3$$

can be restated in the form

$$\boxed{\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c^T x \\ & Ax \geq b \end{aligned}}$$

where
$$A = \begin{pmatrix} A_1 \\ A_2 \\ -A_2 \\ -A_3 \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ -b_2 \\ -b_3 \end{pmatrix},$$

Formulating linear programs

Theorem:
Any linear optimization problem that is given in the form

$$\min_{x \in \mathbb{R}^n} \; c^T x$$
$$Ax \geq b$$

is equivalent to a problem written in *standard form of linear programming*:

$$\boxed{\begin{aligned} &\min_{\tilde{x} \in \mathbb{R}^{2n+m}} \; \tilde{c}^T \tilde{x} \\ &\tilde{A}\,\tilde{x} = b \\ &\tilde{x} \geq 0 \end{aligned}}$$

where

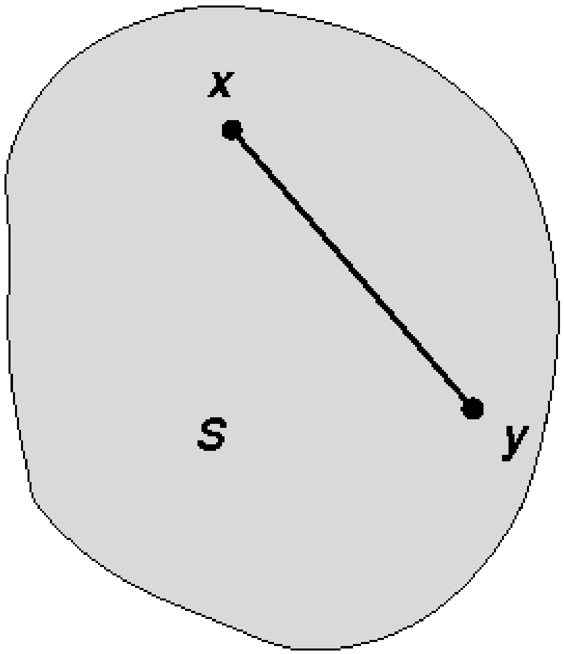
$$\tilde{A} = \begin{pmatrix} A & -A & -I \end{pmatrix}, \qquad \tilde{x} = \begin{pmatrix} x^+ \\ x^- \\ s \end{pmatrix}, \qquad \tilde{c} = \begin{pmatrix} c \\ -c \\ 0 \end{pmatrix}$$

Note: Standard form is more convenient for algorithm development and will therefore frequently be used.

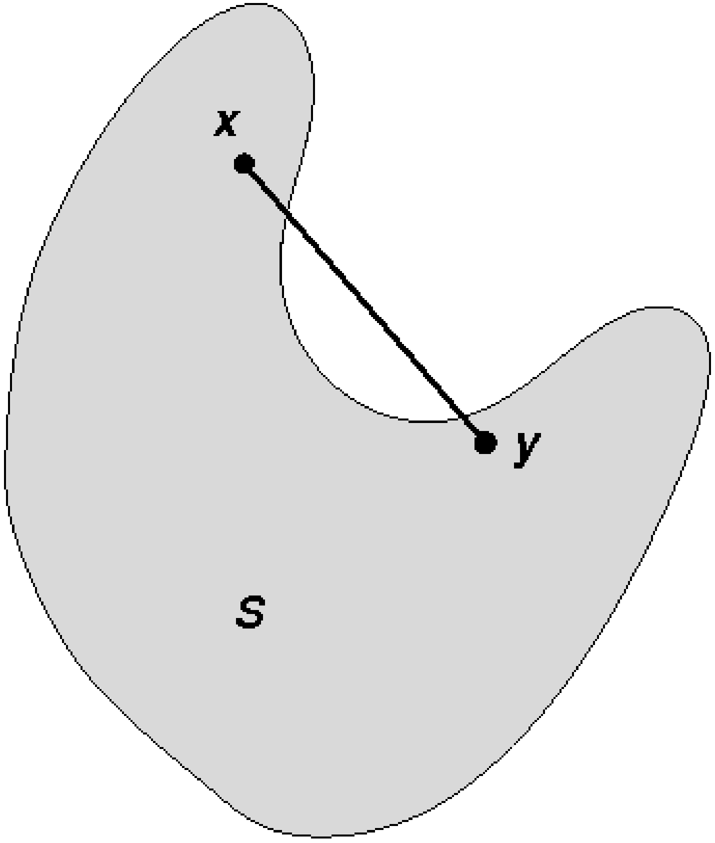
The geometry of feasible sets

Definition: A set S is called convex if

$$x, y \in S \quad \text{implies} \quad \lambda x + (1-\lambda) y \in S \quad \forall\, 0 \le \lambda \le 1.$$



A convex set S.



A nonconvex set S.

The geometry of feasible sets

Lemma: The set of points that satisfy a single constraint,

$$\left\{x \in \mathbb{R}^n : \quad a_i^T x \geq b_i\right\}$$

is a half-space and is convex.

Lemma: The intersection of finitely many convex sets is convex.

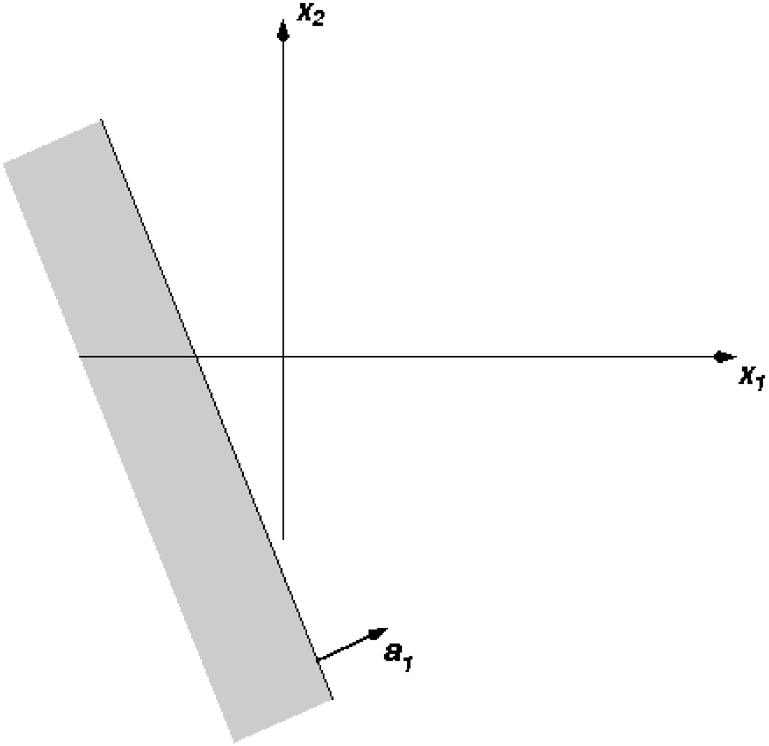
Theorem: The set of points described by the constraints,

$$\left\{x \in \mathbb{R}^n : \quad Ax \geq b\right\}$$

is convex.

The geometry of feasible sets

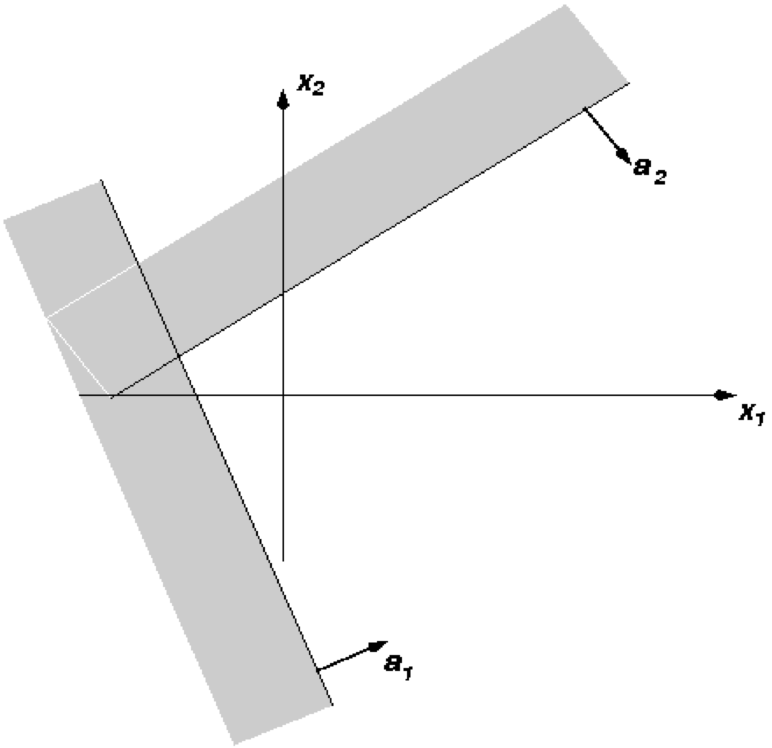
Example: The set of points described by four constraints.



$$\left\{x \in \mathbb{R}^2 : \quad a_i^T x \geq b_i, i = 1\right\}$$

The geometry of feasible sets

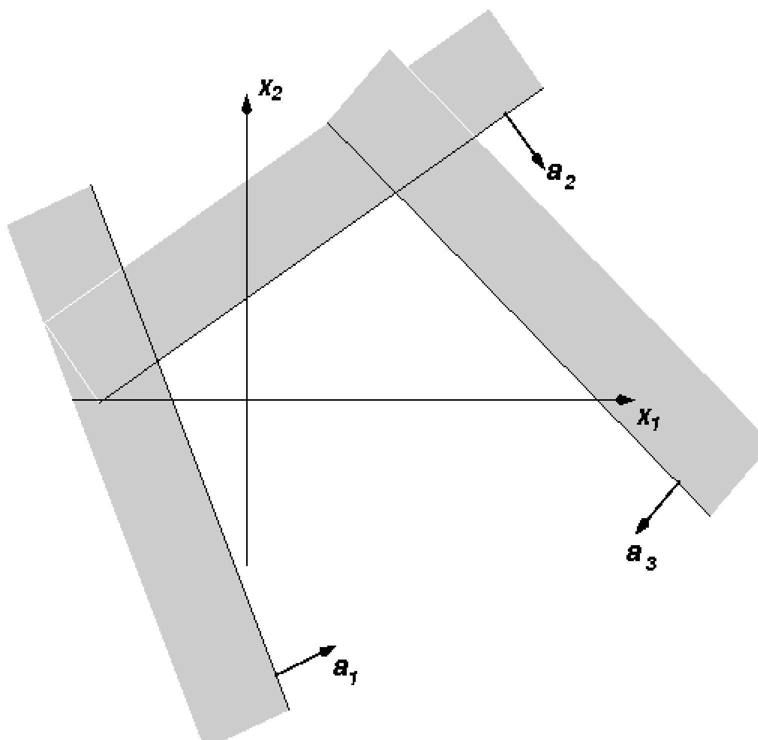
Example: The set of points described by four constraints.



$$\{x \in \mathbb{R}^2 : \ a_i^T x \geq b_i, i = 1,2\}$$

The geometry of feasible sets

Example: The set of points described by four constraints.



Note: The matrix

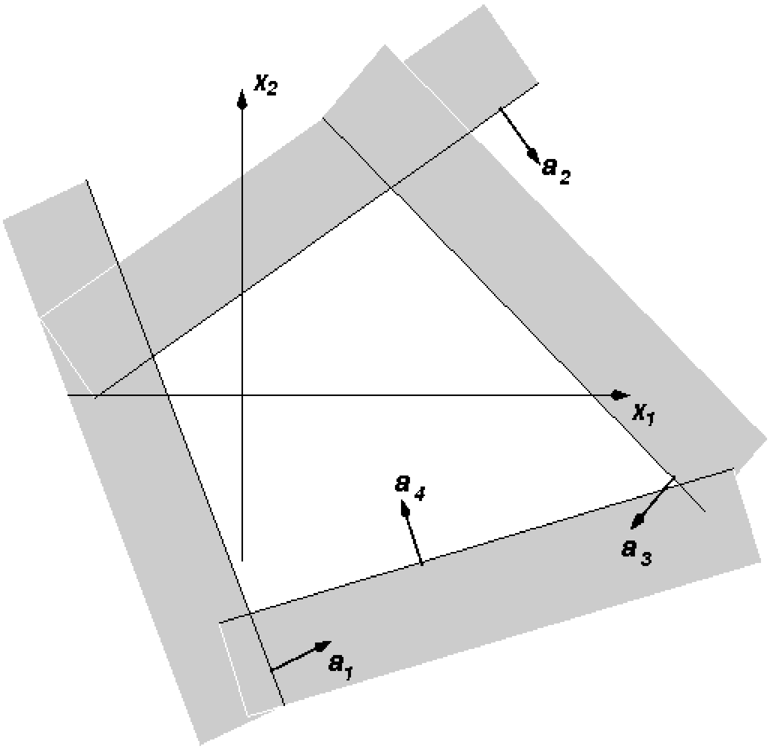
$$A = \begin{pmatrix} a_1^T \\ a_2^T \\ a_3^T \end{pmatrix} \in \mathbb{R}^{3\times 2}$$

does now no longer have full row rank!

$$\{x \in \mathbb{R}^2 : \quad a_i^T x \geq b_i, i = 1..3\}$$

The geometry of feasible sets

Example: The set of points described by four constraints.

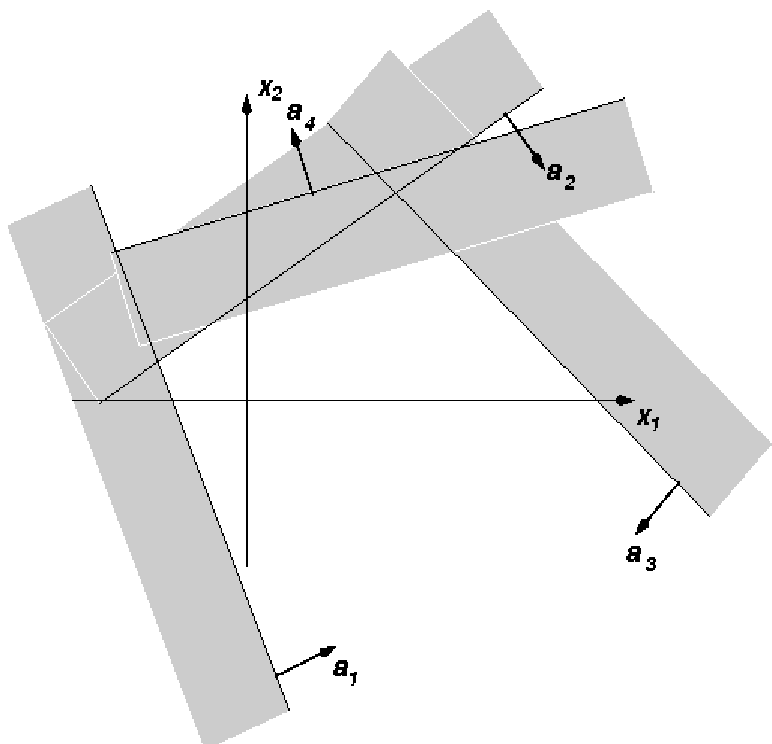


Note: The feasible set is a compact subset of R^2.

$$\{x \in \mathbb{R}^2 : \ a_i^T x \geq b_i, i = 1..4\}$$

The geometry of feasible sets – Pathologies 1

Example: The set of points described by four constraints.

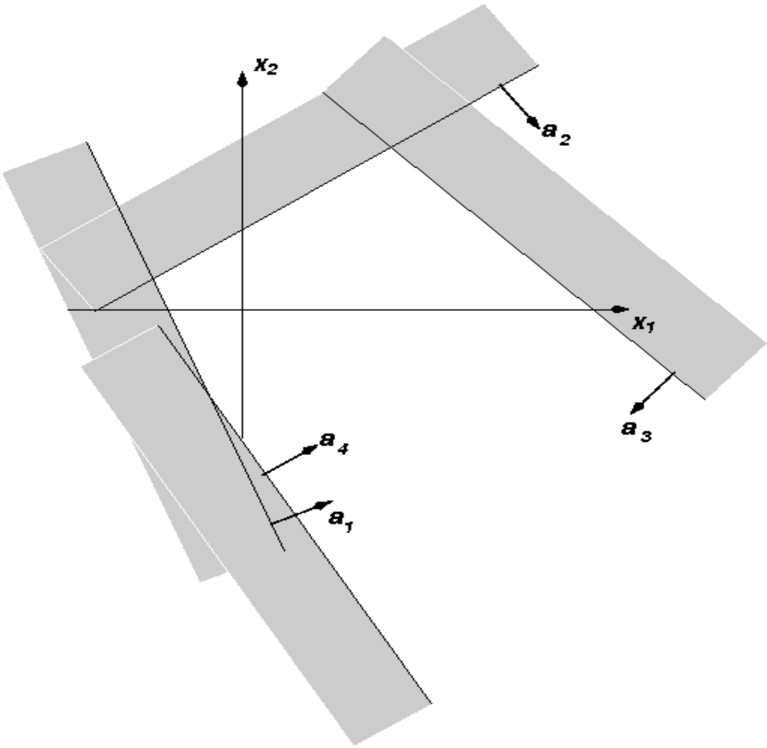


Note: The feasible set is empty. The constraints are said to be *mutually incompatible*!

$$\{x \in \mathbb{R}^2 : \quad a_i^T x \geq b_i, i = 1..4\}$$

The geometry of feasible sets – Pathologies 2

Example: The set of points described by four constraints.

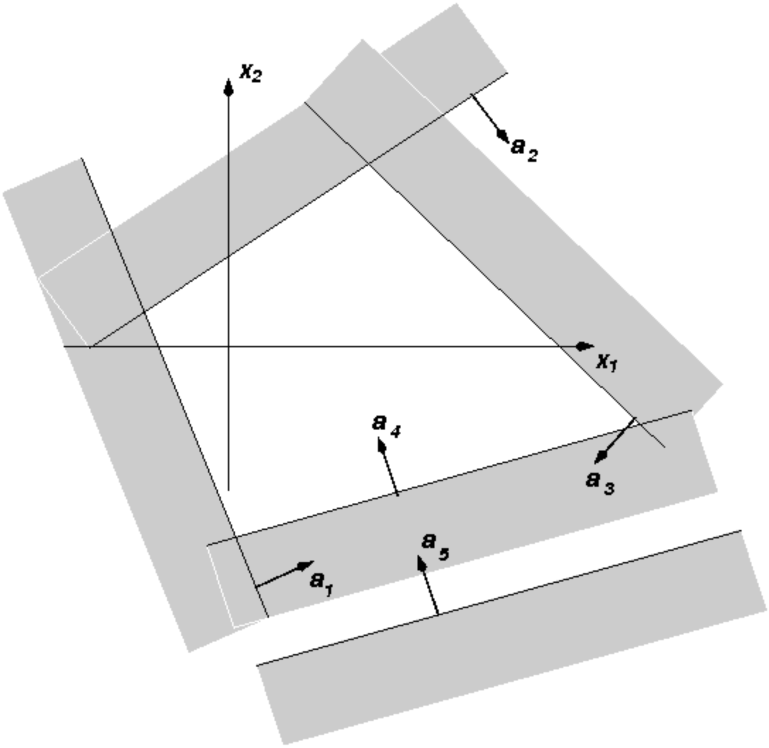


Note: The feasible set is unbounded and consequently not compact.

$$\{x \in \mathbb{R}^2 : \quad a_i^T x \geq b_i, i = 1..4\}$$

The geometry of feasible sets – Pathologies 3

Example: The set of points described by five constraints.

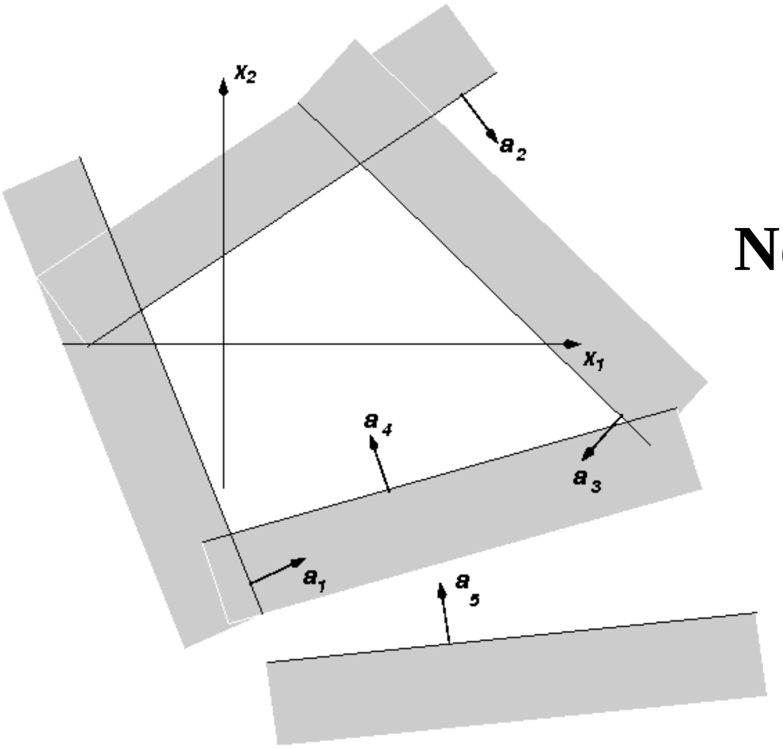


Note: Constraints 4 and 5 are linearly dependent but not mutually exclusive. Nothing bad happens.

$$\{x \in \mathbb{R}^2: \quad a_i^T x \geq b_i, i = 1..5\}$$

The geometry of feasible sets – Pathologies 4

Example: The set of points described by five constraints.

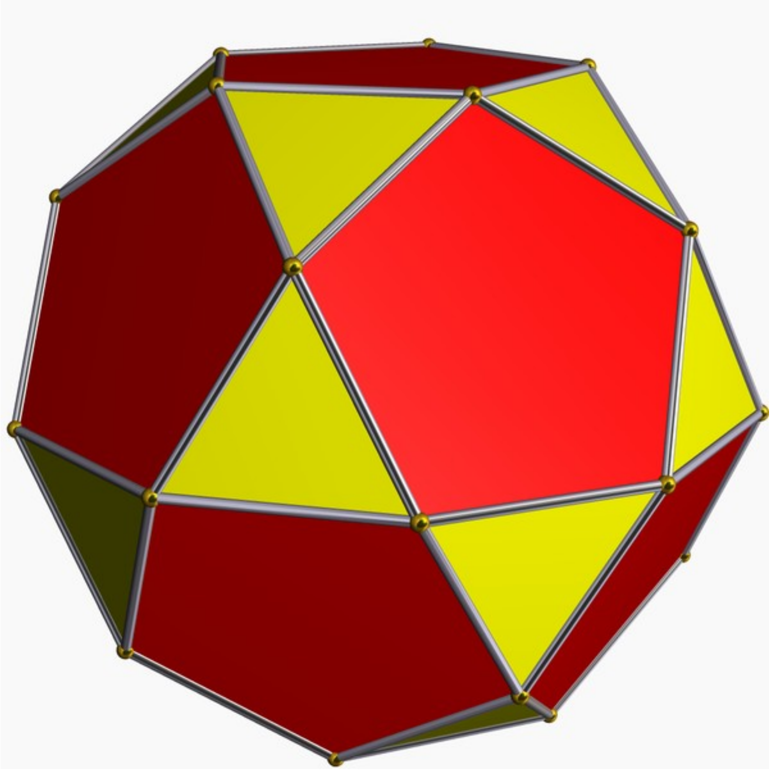


Note: Constraint 5 is not parallel to any of the other constraints but will never be active. Nothing bad happens.

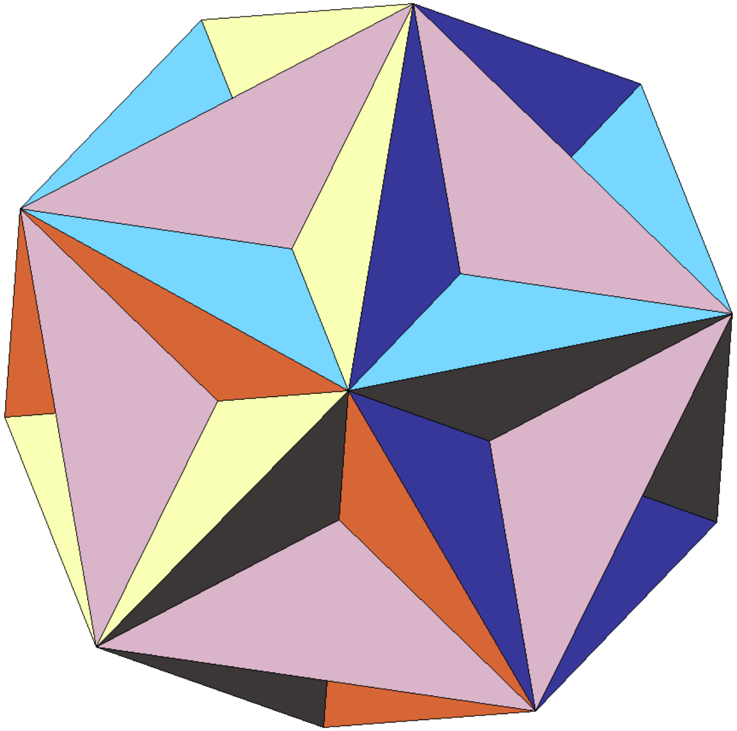
$$\{x \in \mathbb{R}^2 : \; a_i^T x \geq b_i, i = 1..5\}$$

The geometry of feasible sets – 3 and more dimensions

Feasible sets with three or more variables must still be convex. They are, in particular, convex polyhedra:



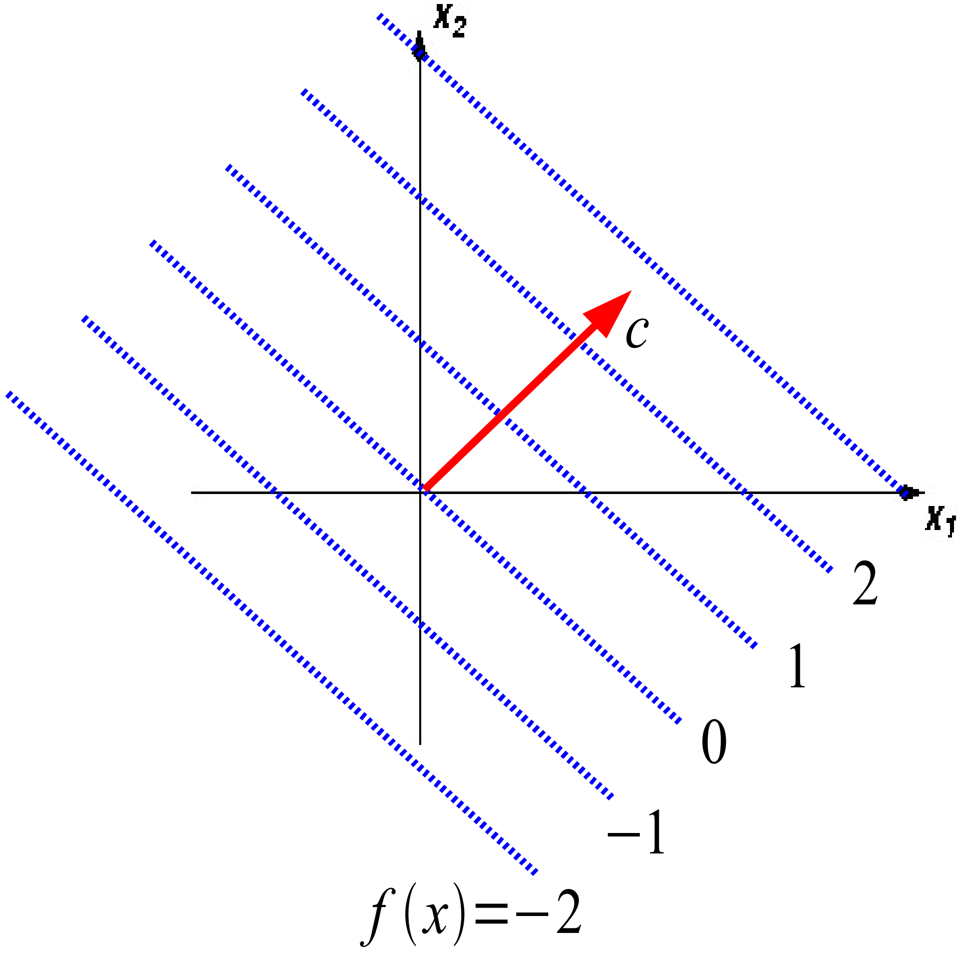
A convex polyhedron
(an icosidodecahedron)



A nonconvex polyhedron
(a stellation)

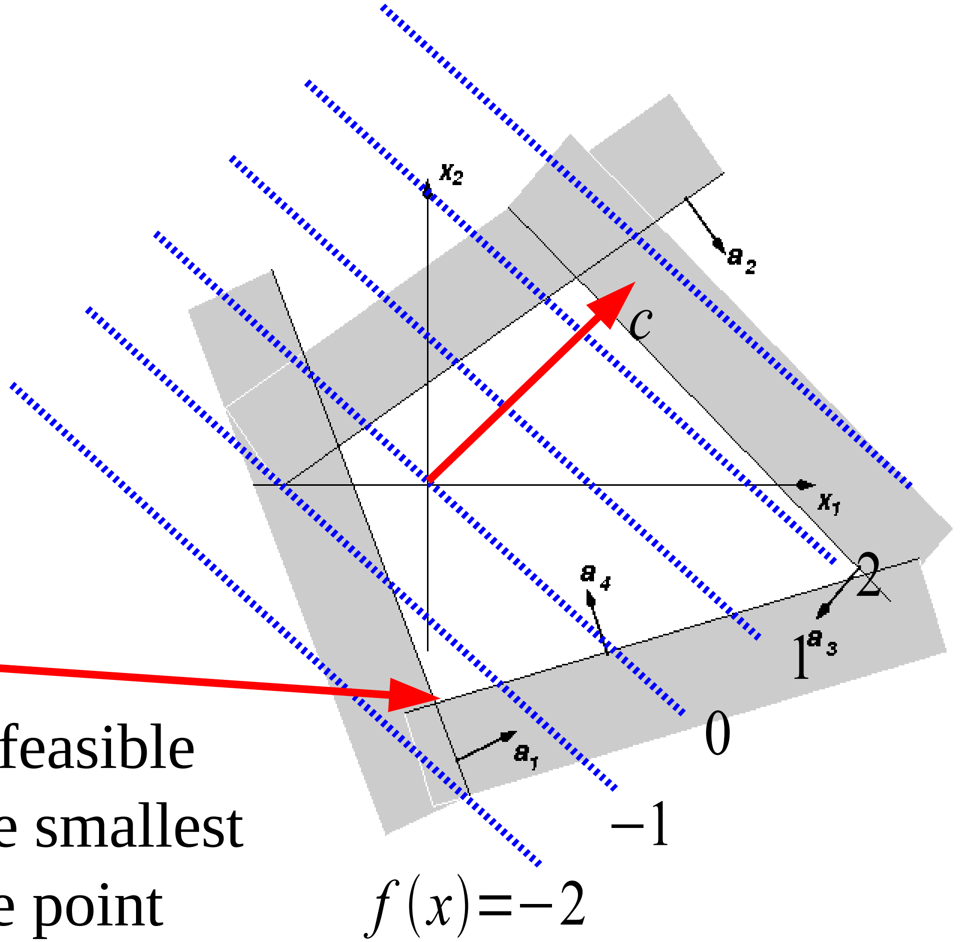
The geometry of the objective function

Example: The function $f(x) = x_1 + x_2 = c^T x$.



The geometry of linear problems

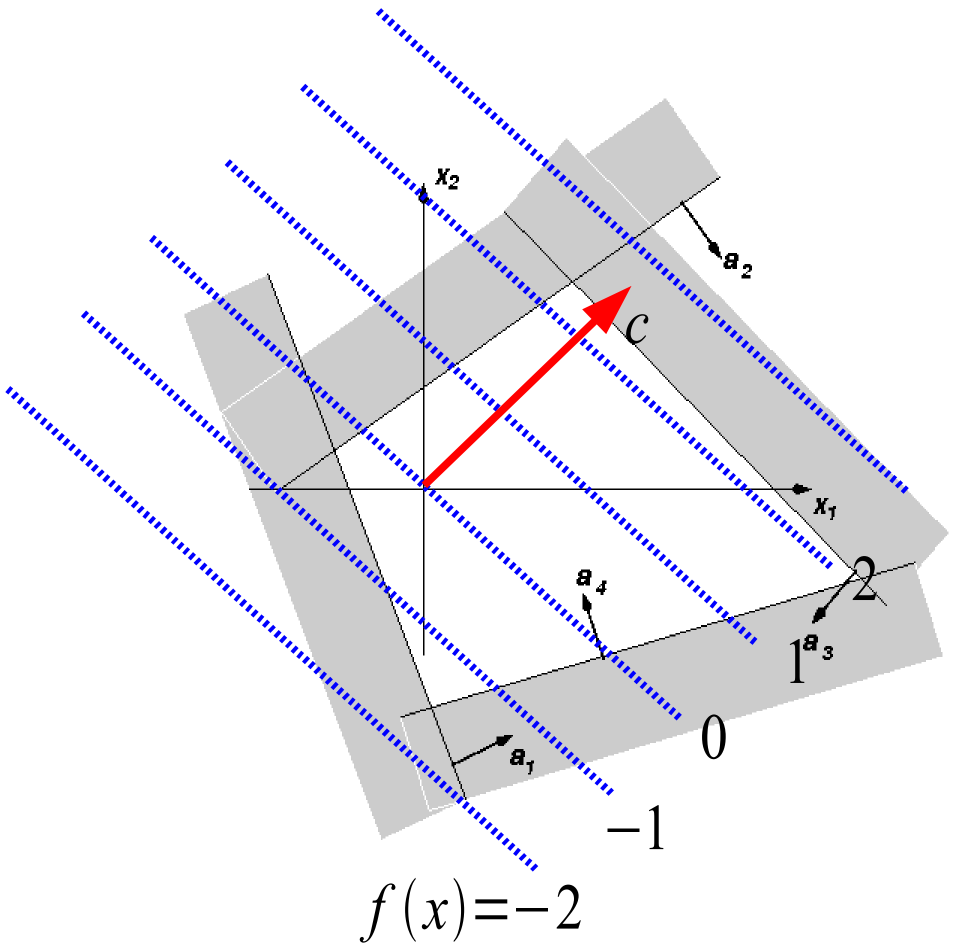
Example: The function $f(x) = x_1 + x_2 = c^T x$.



This is the location within the feasible set with the smallest *f(x)*, i.e. the point of solution.

The geometry of linear problems

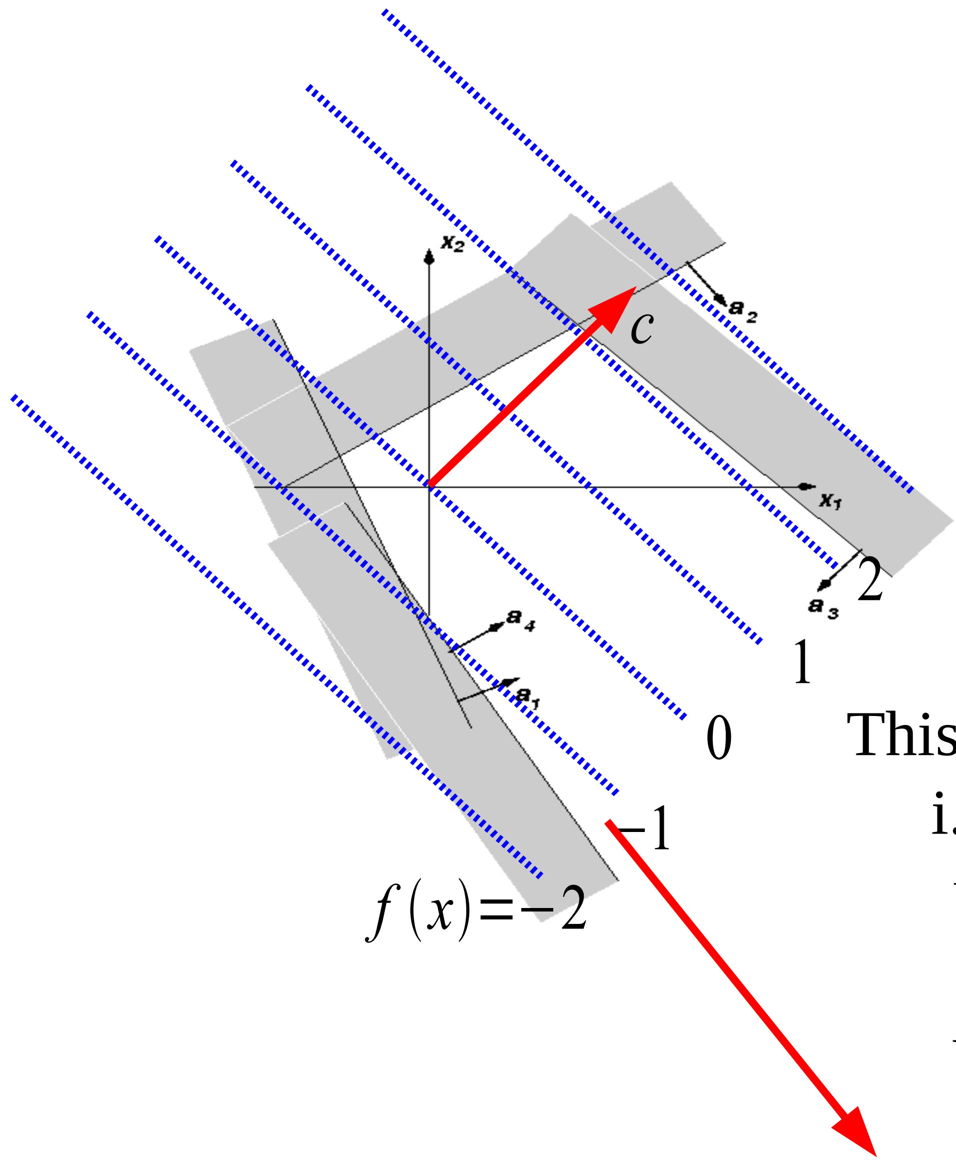
Example: The function $f(x) = x_1 + x_2 = c^T x$.



The geometry of such problem suggests an algorithm: Start at one of the vertices and keep trying to find an adjacent vertex with smaller *f(x)*. This is, in essence, Dantzig's *simplex algorithm*.

The geometry of linear problems

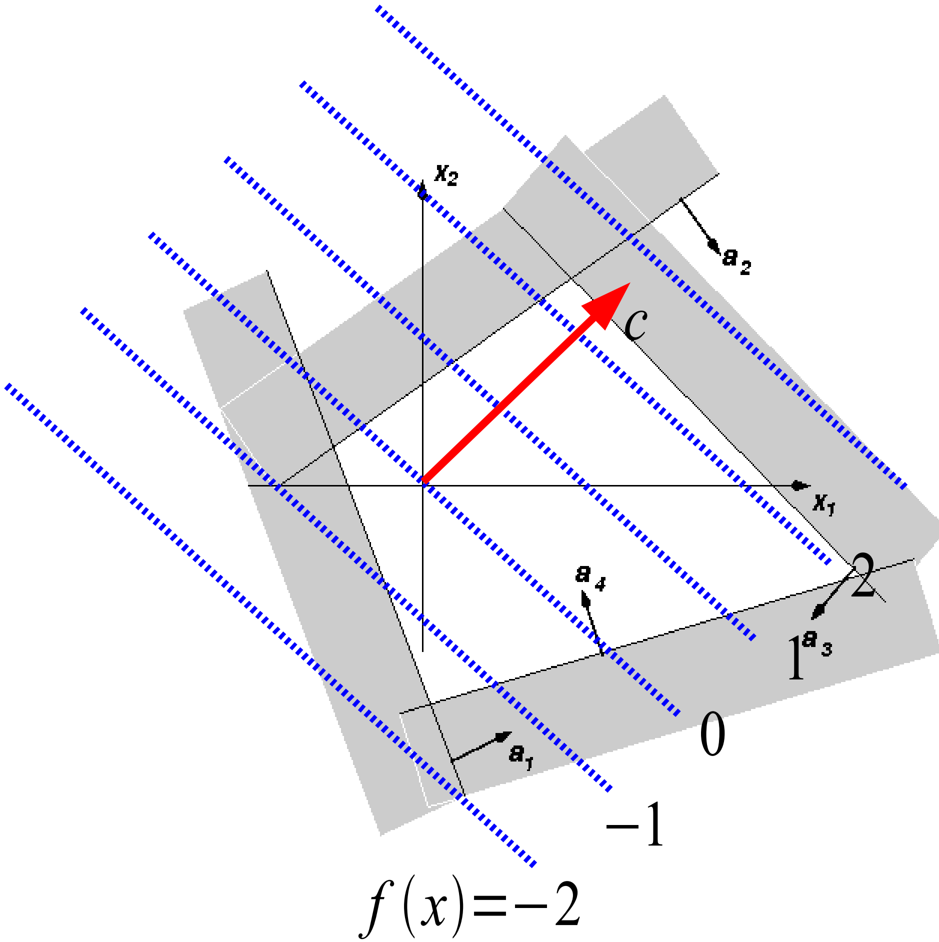
Example: The function $f(x) = x_1 + x_2 = c^T x$.



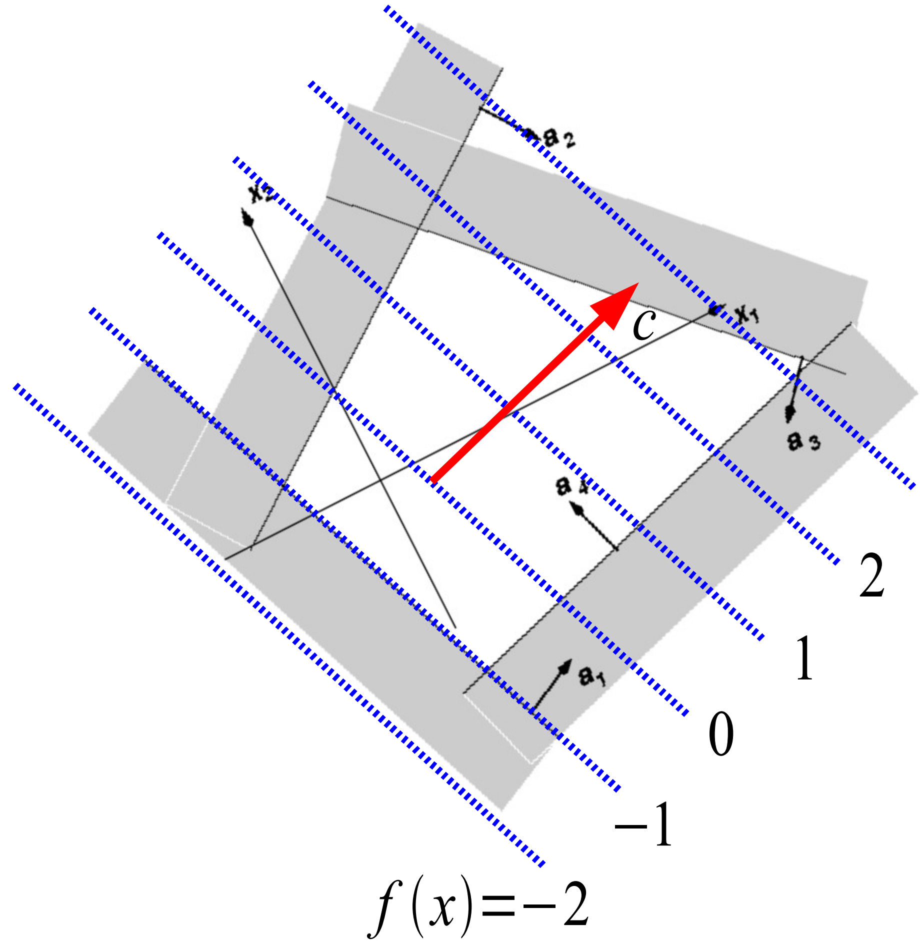
This problem is unbounded, i.e. there is a direction in which the feasible set is unbounded and *f(x)* is unbounded from below.

Possible solutions of linear programs

If (i) the feasible set is bounded, and (ii) the feasible set is not empty:



A single vertex is the unique solution.



All points along a whole edge are solutions. In particular, the vertices of the edge are solutions.

The geometry of linear programs in standard form

Recall that any linear optimization problem that is given in the form

$$\min_{x \in \mathbb{R}^n} \; c^T x$$
$$Ax \geq b$$

is equivalent to a problem written in *standard form*:

$$\min_{\tilde{x} \in \mathbb{R}^{2n+m}} \; \tilde{c}^T \tilde{x}$$
$$\tilde{A}\tilde{x} = b$$
$$\tilde{x} \geq 0$$

where

$$\tilde{A} = \begin{pmatrix} A & -A & -I \end{pmatrix}, \qquad \tilde{x} = \begin{pmatrix} x^+ \\ x^- \\ s \end{pmatrix}, \qquad \tilde{c} = \begin{pmatrix} c \\ -c \\ 0 \end{pmatrix}$$

Note: If $A \in \mathbb{R}^{m \times n}$, then $\tilde{A} \in \mathbb{R}^{m \times (2n+m)}$. While the original matrix may not have fewer rows than columns, the second definitely does.

The geometry of linear programs in standard form

Problems written in *standard form*:

$$\begin{aligned} \min_{\tilde{x} \in \mathbb{R}^{2n+m}} \quad & \tilde{c}^T \tilde{x} \\ & \tilde{A}\tilde{x} = b \\ & \tilde{x} \geq 0 \end{aligned}$$

definitely have a matrix with fewer rows (m) than columns ($2n+m$).

Corollary: The feasible set is the intersection of a hyperplane (with dimension equal to at least $(2n+m)-m=2n$) with the first quadrant/ octant/etc.

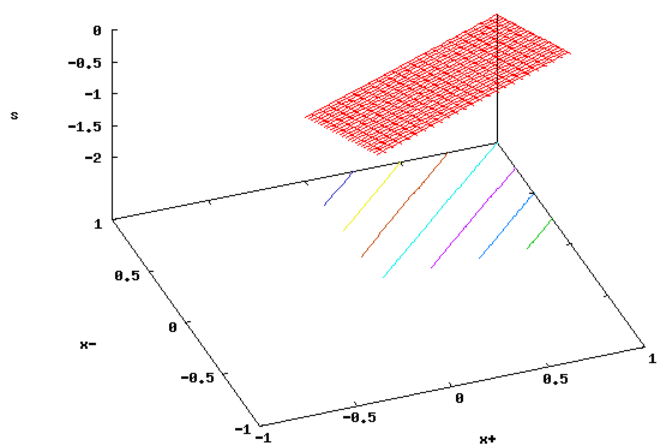
In particular, the feasible set is also a polygon, just like before, except that this polygon now lies in a lower-dimensional subspace defined by the constraint.

The geometry of linear programs in standard form

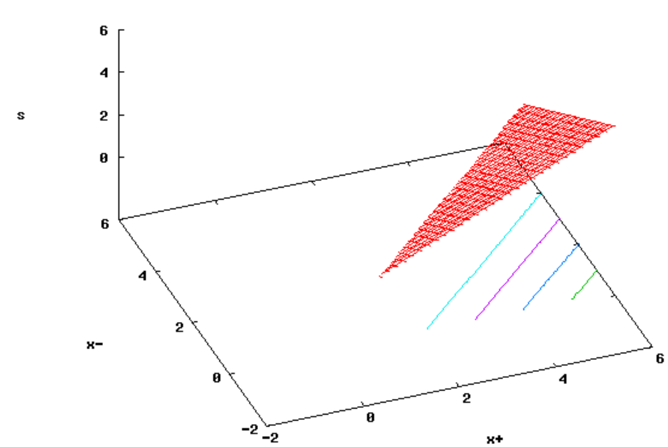
Example: Consider
$$\min_{x \in \mathbb{R}} \quad x$$
$$x \geq 1$$

The standard form of this problem is:

$$\min_{\tilde{x}=\{x^+, x^-, s\} \in \mathbb{R}^3} \quad x^+ - x^-$$
$$x^+ - x^- - s = 1$$
$$\tilde{x} \geq 0$$



$x^+ - x^- - s = 1$ in the area $x^+ \geq 0, x^- \geq 0$



$x^+ - x^- - s = 1$ in the area $\tilde{x} \geq 0$

i.e. the feasible set

The geometry of linear programs in standard form

Example: Consider
$$\min_{x \in \mathbb{R}} \; x$$
$$x \geq 1$$

The standard form of this problem is:

$$\min_{\tilde{x}=\{x^+, x^-, s\} \in \mathbb{R}^3} \; x^+ - x^-$$
$$x^+ - x^- - s = 1$$
$$\tilde{x} \geq 0$$

Note: The solution to this problem is not unique – any set of variables

$$x^+ = 1 + x^-, \quad x^- \geq 0, \quad s = 0$$

produces the optimal value 1 of the objective function. By unsubstituting variables we get the unique solution of the original problem:

$$x = x^+ - x^- = 1$$

The value of the objective function is of course the same.

Possible solutions of linear programs

One of the following cases must hold:

- A vertex of the feasible region is the unique solution
- All points of an edge or face of the feasible region are solutions; in particular, the vertices of the edge or face are solutions
- The feasible set is empty and there are no solutions
- The feasible set is unbounded and the objective function is unbounded from below in one of the directions in which the feasible set is unbounded; the problem then has no bounded solution.

In other words:

If bounded solutions exists,
the set of solutions must include at least one vertex!

Possible solutions of linear programs

Definition:

We call a point x^* a local solution of

$$\begin{aligned}\min_{x\in\mathbb{R}^n}\ & f(x)\\ & g(x)=0,\ \ h(x)\ge 0\end{aligned}$$

if there exists a neighborhood U of x^* so that

$$f(x^*)\le f(x)\quad \forall x\in U\cap\{x: g(x)=0,\ h(x)\ge 0\}$$

Definition:

We call a point x^* a global solution of

$$\begin{aligned}\min_{x\in\mathbb{R}^n}\ & f(x)\\ & g(x)=0,\ \ h(x)\ge 0\end{aligned}$$

if

$$f(x^*)\le f(x)\quad \forall x\in\{x: g(x)=0,\ h(x)\ge 0\}$$

Possible solutions of linear programs

Definition:

We call a function *f(x)* convex if

$$f(\lambda x+(1-\lambda)y) \leq \lambda f(x)+(1-\lambda)f(y) \quad \forall x, y \in D \subset \mathbb{R}^n, \lambda \in [0,1]$$

We call it concave if

$$f(\lambda x+(1-\lambda)y) \geq \lambda f(x)+(1-\lambda)f(y) \quad \forall x, y \in D \subset \mathbb{R}^n, \lambda \in [0,1]$$

Corollary:

Any linear (affine) function is both convex and concave.

Remark: In fact, affine functions are the only functions that are both convex and concave.

Possible solutions of linear programs

Theorem:
Any local solution of a linear program is also a global solution.

(Proof: Use convexity of the feasible set and of the objective function.)

Theorem:
The set of all global solutions of a linear program is convex (and consequently also singly connected).

(Proof: Use convexity of the feasible set and linearity of the objective function.)

Theorem:
Among all solutions of a linear program is always at least one vertex of the feasible set.

(Proof: Later, need to define precisely what a vertex is.)

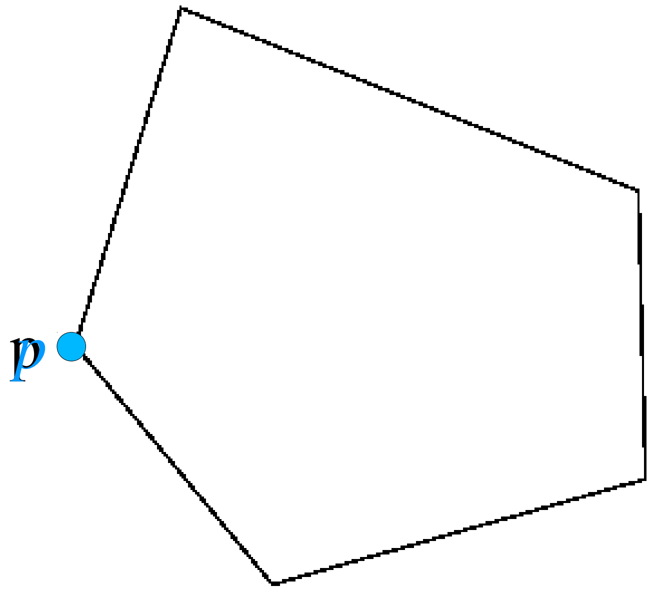
Polyhedra in a formal language

Definition:
We call the set of points $\{x \in \mathbb{R}^n : Ax \geq b\}$ a *polyhedron*.
If *n=2*, we also call it a *polygon.*

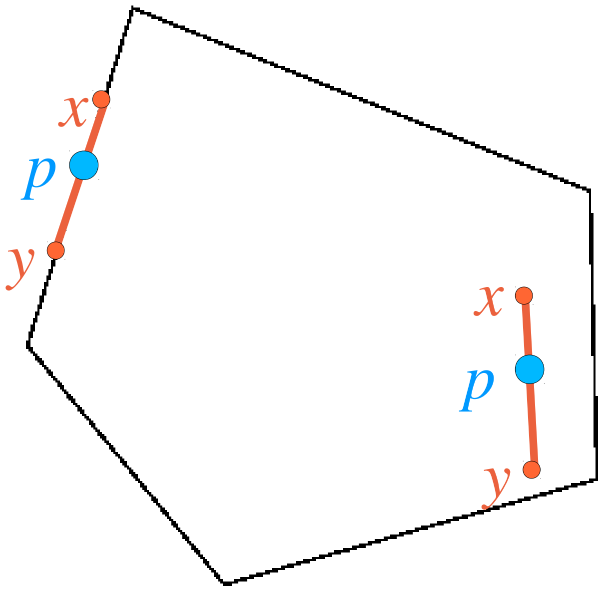
Corollary:
The set of points $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ is also a polyhedron.

Polyhedra in a formal language

Definition:
Let $P \subset \mathbb{R}^n$ be a polyhedron. We call $p \in P$ an *extreme point* if there are no $x, y \in P, x \neq p, y \neq p$ so that $p = \lambda x + (1-\lambda) y$ for any $0 \leq \lambda \leq 1$.



An extreme point.



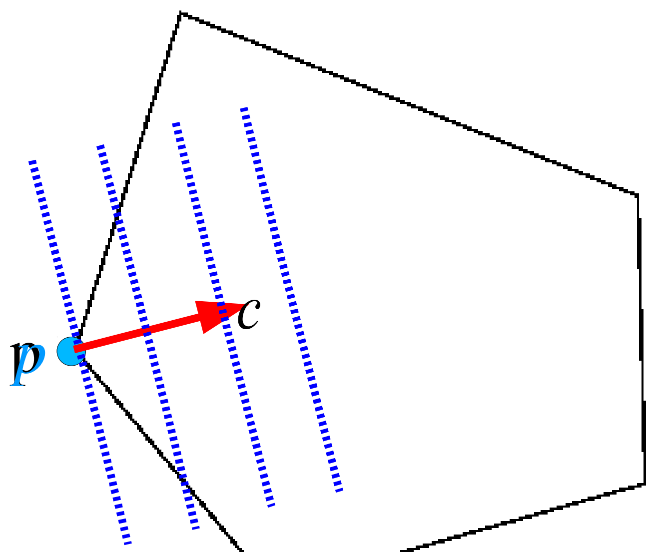
Two points that are not extreme.

Polyhedra in a formal language

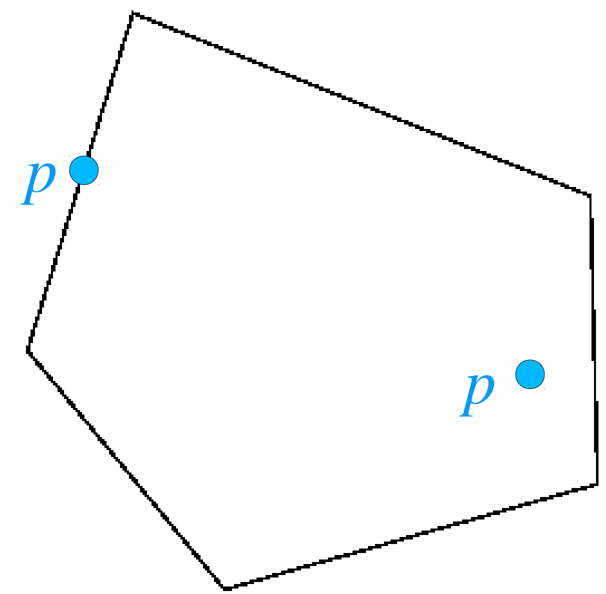
Definition:

Let $P \subset \mathbb{R}^n$ be a polyhedron. We call $p \in P$ a *vertex* of P if there is a vector c so that

$$c^T p < c^T x \quad \forall x \in P, x \neq p.$$



An extreme point.



Two points that are not extreme.

Polyhedra in a formal language

Definition:
Let $P \subset \mathbb{R}^n$ be a polyhedron defined by

$$P = \left\{ x \in \mathbb{R}^n : a_i^T x \geq b_i \text{ for } i = 1 \ldots m_1, \quad a_i^T x = b_i \text{ for } i = m_1 + 1 \ldots m_2 \right\}$$

The set of *active* or *binding* constraints at a arbitrary point $p \in \mathbb{R}^n$ is defined as

$$I(p) = \left\{ i \in [1, m_1] : a_i^T p = b_i \right\} \cup \left\{ i \in [m_1 + 1, m_2] : a_i^T p = b_i \right\} \subset [1, m_2]$$

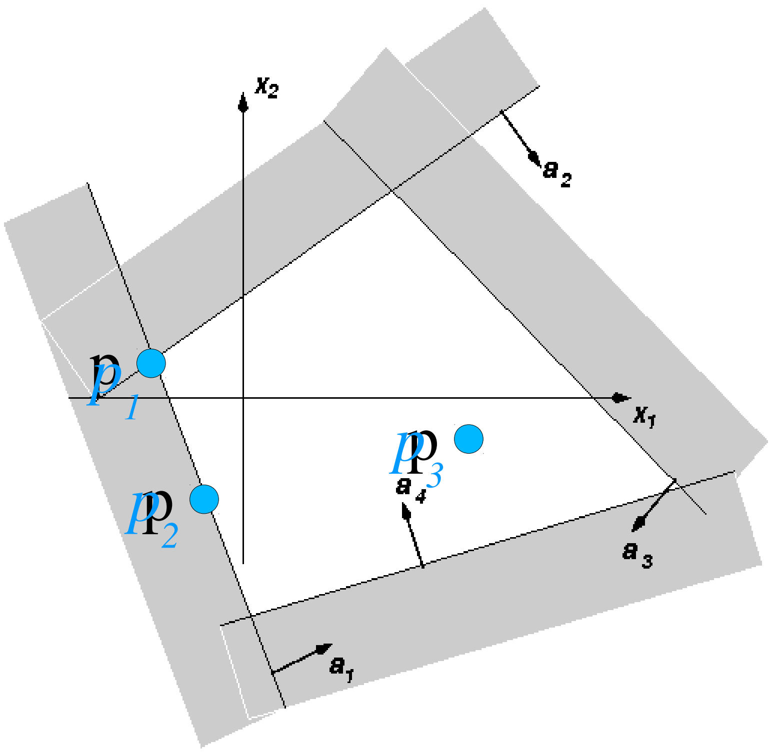
Note: If $p \in P$ then

$$I(p) = \left\{ i \in [1, m_1] : a_i^T p = b_i \right\} \cup [m_1 + 1, m_2]$$

*b*ecause all equality constraints must be active.

Polyhedra in a formal language

Example:



$$I(p_1)=\{1,2\}$$

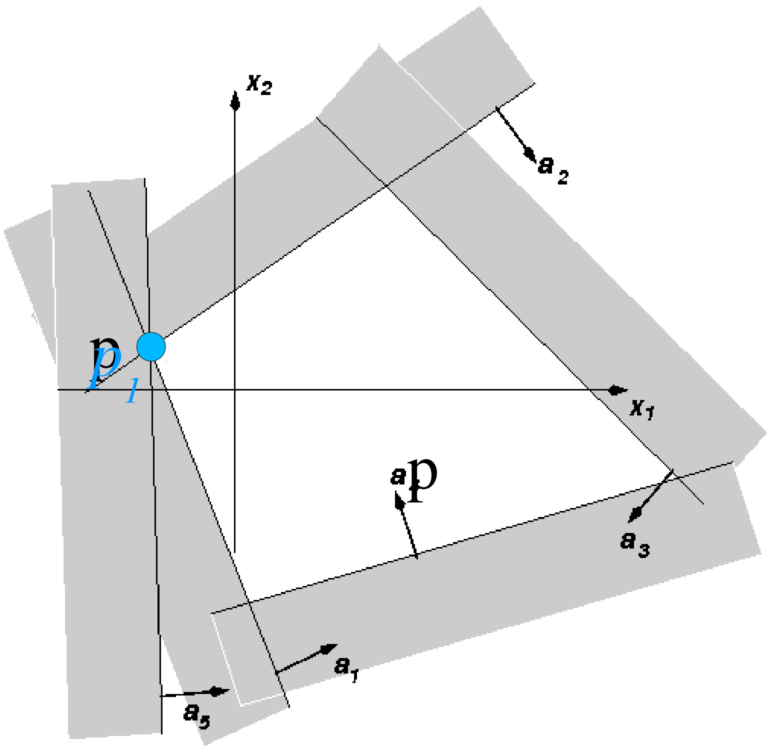
$$I(p_1)=\{1\}$$

$$I(p_1)=\{\}$$

Tentative conclusion: At a vertex of a polyhedron in *n* space dimension, *n* constraints are active, i.e. $\#I(p)=n$.

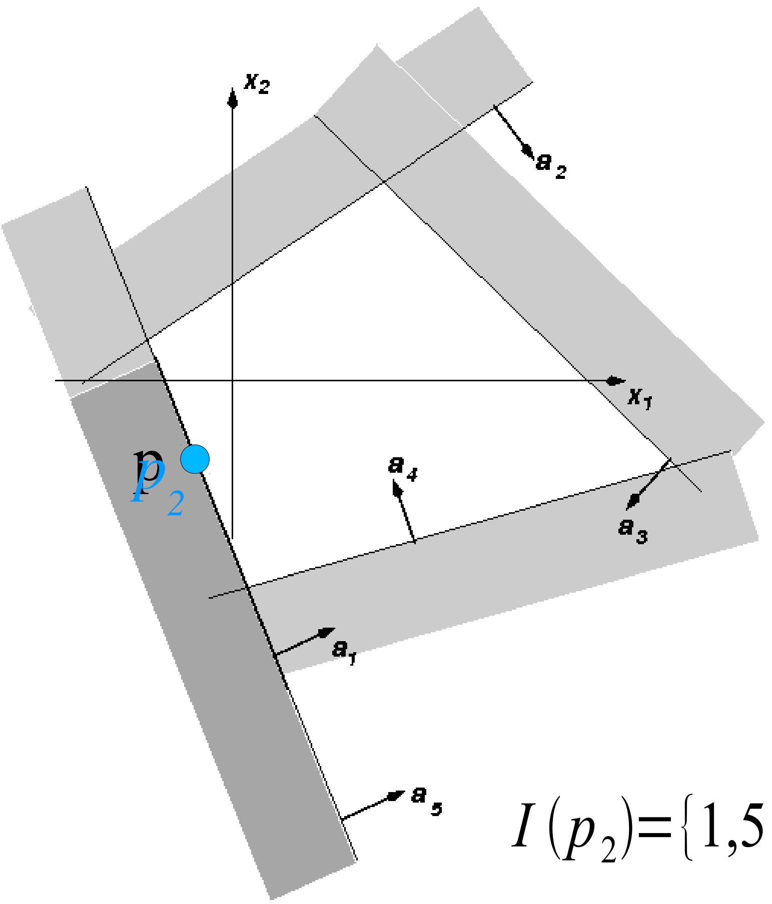
Polyhedra in a formal language

But careful:



$I(p_1)=\{1,2,5\}$

A vertex with *n+1* active constraints.



$I(p_2)=\{1,5\}$

Not a vertex, but *n* constraints are active.

Polyhedra in a formal language

Definition:

Let $P \subset \mathbb{R}^n$ be a polyhedron defined by

$$P = \left\{ x \in \mathbb{R}^n : a_i^T x \geq b_i \text{ for } i = 1 \ldots m_1, \quad a_i^T x = b_i \text{ for } i = m_1 + 1 \ldots m_2 \right\}$$

We call $p \in \mathbb{R}^n$ a *basic solution of P* if:

- all equality constraints are satisfied at *p*
- the set

  $$\left\{ a_i : i \in I(p) \right\}$$

  contains *n* vectors that are linearly independent.

Polyhedra in a formal language

Note:
The condition that

$$\{a_i : i \in I(p)\}$$

contains *n* vectors that are linearly independent is equivalent to saying that

- The vectors a_i form a *basis* of R^n. This is why these points are called "basic" (i.e. "basic" as in "basis", not "fundamental").

- If we group the vectors and corresponding right hand sides into a linear system

$$a_i^T x = b_i, \qquad i \in I(p)$$

then the solution will be unique and will equal *p*. This is why these points are called "solutions".

Polyhedra in a formal language

Definition:

Let $P \subset \mathbb{R}^n$ be a polyhedron defined by

$$P = \left\{ x \in \mathbb{R}^n : a_i^T x \geq b_i \text{ for } i = 1 \ldots m_1, \quad a_i^T x = b_i \text{ for } i = m_1 + 1 \ldots m_2 \right\}$$

We call $p \in \mathbb{R}^n$ a *degenerate basic solution of P* if:

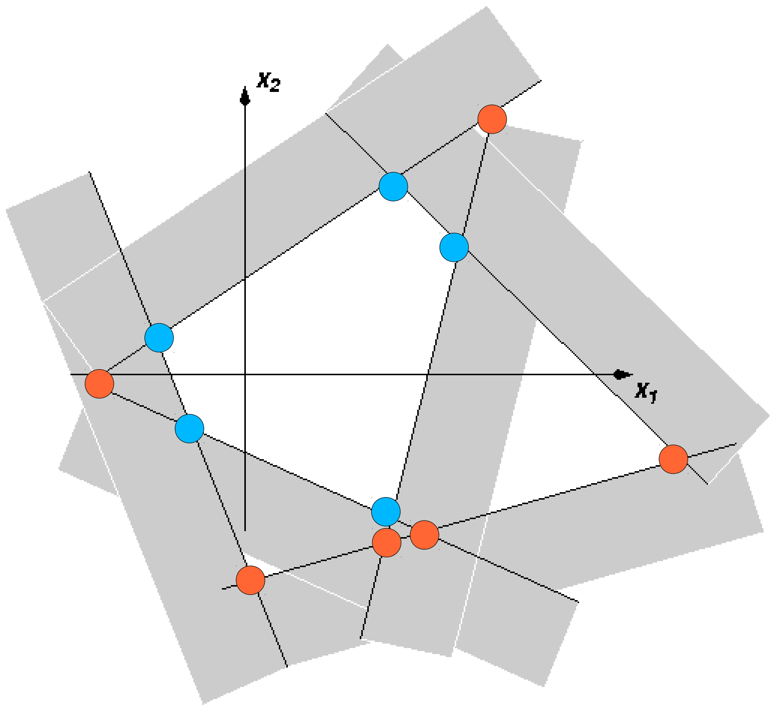
- all equality constraints are satisfied at *p*
- the set

  $$\left\{ a_i : i \in I(p) \right\}$$

  contains *n* vectors that are linearly independent
- this set has more than *n* elements, i.e. more than *n* constraints are active at *p*.

Polyhedra in a formal language

Definition:
A basic solution p is called a *feasible basic solution* if in addition to the equality and active inequality constraints also the inactive inequality constraints are satisfied.



- The only five feasible basic solutions
- Some of the non-feasible basic solutions

Polyhedra in a formal language

Theorem:
A feasible basic solution is a vertex is an extreme point.

In other words: Let *p* be a point in a non-empty polyhedron *P*. Then it is either none or all of the following:

- *p* is a feasible basic solution
- *p* is a vertex of *P*
- *p* is an extreme point of *P*

Polyhedra in a formal language

Theorem:
A polyhedron can only have finitely many vertices.

Note: In fact, a polyhedron

$$P = \{ x \in \mathbb{R}^n : a_i^T x \geq b_i \text{ for } i = 1 \ldots m', \quad a_i^T x = b_i \text{ for } i = m'+1 \ldots m \}$$

can have at most

$$\binom{m}{n} = \frac{m!}{(m-n)!\,n!}$$

basic solutions. However, this can be a very large number!

Example: The unit cube in *n* dimensions has $2^n \approx 10^{0.3n}$ vertices.

Polyhedra in a formal language

Note: A polyhedron

$$P=\{x\in\mathbb{R}^n : a_i^T x\geq b_i \text{ for } i=1...m', \quad a_i^T x=b_i \text{ for } i=m'+1...m\}$$

can have at most

$$\binom{m}{n}=\frac{m!}{(m-n)!\,n!}$$

basic solutions. However, not all of them are feasible. In fact, at every feasible basic solution, the *m-m'* equality constraints need to all be active, so that there can be at most

$$\binom{m'}{n-(m-m')}=\frac{m'!}{(n-m+m')!\,(m'-(n-(m-m')))!}=\frac{m'!}{(n-(m-m'))!\,(m-n)!}$$

feasible basic solutions (vertices).

Possible solutions of linear programs

Theorem:
If the feasible set of a linear program is non-empty and has at least one vertex, then exactly one of the following is true:

- The minimum of the objective function over the feasible set is $-\infty$, or
- among all solutions of the linear program is always at least one vertex of the feasible set.