What happens for inequality constraints?

⋆ minimize $f(x)$ such that $g_1(x) \leq 0, ..., g_k(x) \leq 0$.
⋆ not all inequality constraints play the same role: at the point x the constraint i is said to be active if $g_i(x) = 0$.
⋆ if a constraint g_i (where g_i is C^1) is inactive at a minimizer x^* then $g_i(x) < 0$ in a neighborhood of x^*

⋆ if x^* is a minimizer of $f(x)$ under the constraints g_i and $g_i(x^*) < 0$ then g_i does not impose any restriction on f locally: ignoring it produces the same result (locally)
⋆ equality constraints generally produced surfaces while inequality constraints can just give bunded regions of $\mathbb{R}^n$.

Qualification of constraints

⋆ denote by $I(x) = \{i \in \{1, ..., k\} : g_i(x) = 0\}$ be the indices of active constraints at x

⋆ we say that the constraints are qualified at x if the gradients $(\nabla g_i(x))_{i\in I(x)}$ are linearly independent!

⋆ geometrically, as in the equality constraints case, if the constraints are qualified at x then we may define a proper tangent space using the family $(\nabla g_i(x))_{i\in I(x)}$

⋆ **Special case:** if all g_i are affine constraints then they are automatically qualified. Why?

- in this case the constraints also define the tangent space themselves
- the linear independence of the gradients at a point x is equivalent to the removal of redundant constraints

Optimality conditions: inequalities

Theorem 4

Let x^* be a local minimizer for the inequality constrained problem

$$\min_{g(x)\leq 0} f(x)$$

and suppose that the constraints are qualified at x^*. Then the following affirmations are true:

- *There exists a uniquely defined vector of Lagrange multipliers $\lambda_i^* \geq 0,\ i = 1, ..., k$ such that*

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0.$$

- *Moreover, if $g_i(x^*) < 0$ then $\lambda_i = 0$, also called the complementary slackness relations. Equivalent formulation: $\lambda_i g_i(x^*) = 0$.*

$\star$ why are Lagrange multipliers non-negative in this case? x^* would like to "get out of the constraints" to increase the value of f

$\star$ if x^* is an interior point for $g(x) \leq 0$ then simply $\nabla f(x^*) = 0$

Example: qualification of constraints

Consider the set

$$k = \{x = (x_1, x_2) \in \mathbb{R}^2 : -x_1 \leq 0,\ -x_2 \leq 0,\ -(1 - x_1)^3 + x_2 \leq 0\}.$$

$\star$ Maximize $J(x) = x_1 + x_2$ for $x \in K$.

$\star$ making a drawing we find that immediately that the solutions are $(0, 1)$ and $(1, 0)$.

$\star$ let's check if we can write the optimality condition at the two points:

- $(1, 0)$: constraints not qualified: unable to write the opt. cond
- $(0, 1)$: constraints qualified: the optimality condition can be written!

The Lagrangian - inequality case

$\star$ the optimality conditions obtained involve the gradient of the objective function and the constraints.
$\star$ the optimality condition can be written as the gradient of a function combining the objective and the constraints called the Lagrangian: $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m_+$

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1}^{k} \lambda_i g_i(x) = f(x) + \lambda \cdot g(x).$$

$\star$ if x^* is a local minimum of f on the set $\{g(x) \leq 0\}$ then the optimality condition tells us that there exists $\lambda^* \in \mathbb{R}^m_+$ such that

$$\frac{\partial \mathcal{L}}{\partial x}(x^*, \lambda^*) = 0 \text{ and } \frac{\partial \mathcal{L}}{\partial \lambda}(x^*, \lambda^*) = 0$$

$\star$ moreover, $\sup_{\lambda \in \mathbb{R}^m_+} \mathcal{L}(x, \lambda) = \begin{cases} f(x) & \text{if } g(x) \leq 0 \\ +\infty & \text{otherwise} \end{cases}$ which gives

$$\min_{g(x) \leq 0} f(x) = \min_{x \in \mathbb{R}^n} \sup_{\lambda \in \mathbb{R}^m_+} \mathcal{L}(x, \lambda).$$

$\star$ the minimizer of f becomes a saddle point for the Lagrangian

Come back to the optimal can problem

- ⋆ Area of the can (to be minimized): $A(h,r) = 2\pi r^2 + 2\pi rh$
- ⋆ Volume of the can (constraint): $V(h,r) = \pi r^2 h$
- ⋆ finally we obtain the problem

$$\min_{V(h,r)\geq c} A(h,r).$$

- ⋆ the constraint will be active!
- ⋆ write the optimality condition: find r and h in terms of λ and finish!

⋆ in the end we find that the optimal can will have the height h equal to two times its radius r.

⋆ find now the optimal cup: only one of the two ends is filled with material!

Saddle points

Definition 5

We say that $(u, p) \in U \times P$ is a saddle point of $\mathcal{L}$ on $U \times P$ if

$$\forall q \in P \quad \mathcal{L}(u, q) \leq \mathcal{L}(u, p) \leq \mathcal{L}(v, p) \quad \forall v \in U$$

$\star$ when fixing p: $v \mapsto \mathcal{L}(b, p)$ is minimal for $v = u$

$\star$ when fixing u: $q \mapsto \mathcal{L}(u, q)$ is minimal for $q = p$

$\star$ If J is the objective function and F defines the constraint set K (equality or inequality) then a saddle point (u, p) for the Lagrangian

$$\mathcal{L}(v, q) = J(v) + q \cdot F(v)$$

verifies that u is a minimum of J on K.

$\star$ moreover, if the Lagrangian is defined on an open neighborhood U of the constraint set K then we also recover the optimality condition

$$\nabla J(u) + \sum_{i=1}^{m} p_i \nabla F_i(u) = 0.$$

Sufficient conditions

$\star$ two options: go to the second order or use convexity
$\star$ it is not enough to look at the second order approximation of f on the tangent space! The curvature of the constraint can also play a role.
$\star$ the correct way is to look at the Hessian of the Lagrangian with respect to x, reduced to the tangent space!

$\star$ in the convex case, for inequality constraints things are a little bit easier!

$\star$ why only for inequality constraints? Imagine that equality constraints can produce curved surfaces and the only way to have convexity there is if they are flat!
$\star$ why the choice $g_i(x) \leq 0$ as the definition of inequality constraints? Because if all g_i are convex functions then

$$K = \{x : g_i(x) \leq 0, i = 1, ..., k\} \text{ is a convex set.}$$

Sufficient conditions - convex case

Theorem 6 (Kuhn-Tucker)

Suppose that the functions $f, g_i,\ i = 1, ..., k$ are C^1 and convex. Define K as the set $K = \{x : g_i(x) \le 0\}$ and introduce the Lagrangian

$$\mathcal{L}(v, q) = f(v) + q \cdot g(v),\ v \in \mathbb{R}^n, q \in \mathbb{R}^k_+.$$

Let x^* be a point of K where the constraints are qualified. Then the following are equivalent:

- *x^* is a global minimum of f on K*
- *there exists $\lambda^* \in \mathbb{R}^m$ such that (x^*, λ^*) is a saddle point for the Lagrangian*
- $g(x^*) \le 0$, $\lambda^* \ge 0$, $\lambda^* \cdot F(x^*) = 0$, $\nabla f(x^*) + \sum_{i=1}^k \lambda_i^* \nabla g_i(x^*) = 0$.

$\star$ why the reverse implication works? When $q \ge 0$ the Lagrangian

$$\mathcal{L}(v, q) = f(v) + q \cdot g(v),\ v \in \mathbb{R}^n, q \in \mathbb{R}^k_+$$

is convex when f and $g = (g_i)$ are convex!

$\star$ particular case: affine equalities! convex and qualified!

Handle the constraints numerically

⋆ we already saw two methods:

- projected gradient algorithm:
$$x_{i+1} = \text{Proj}_K(x_i - t\nabla f(x_i))$$
- penalization: include the constraint $\{g = 0\}$ in the objective
$$\min f(x) + \frac{1}{\varepsilon} g(x)^2$$

⋆ we saw that the projection is not explicit in most cases! In the meantime we learned how to solve non-linear equations. Imagine the following algorithm:

- Compute x_i and the projection d_i of $-\nabla f(x_i)$ on the tangent space (orthogonal of $(\nabla g_j(x_i))$)
- advance in the direction of d_i: $x_{i+1} = x_i + \gamma_i d_i$
- project x_{i+1} on the set of constraints using the Newton method

Conclusion on Lagrange multipliers

- we may obtain necessary optimality conditions involving equality and inequality constraints: the gradient of f is a linear combination of the gradients of the constraints
- the gradients of the constraints need to be linearly independent at the optimum: proper definition of the tangent space!
- for inequality constraints only the active constraints come into play in the optimality condition
- sufficient conditions can be found in the convex case: Kuhn-Tucker theorem
- the theory gives new ways to handle constraints numerically

Constrained optimization

- General theoretical and practical aspects
- A quick intro to linear programming

Linear programming

⋆ maximizing or minimizing a linear function subject to linear constraints!

⋆ Example:

$$\max(x_1 + x_2)$$

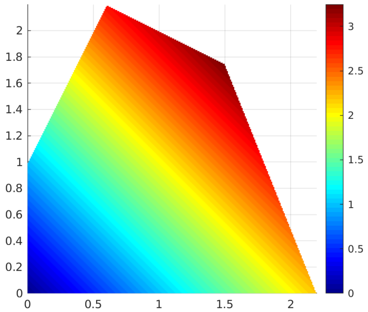
such that $x_1 \geq 0,\ x_2 \geq 0$ and

$$\begin{array}{rcl} x_1 + 2x_2 & \leq & 5 \\ 5x_1 + 2x_2 & \leq & 11 \\ -2x_1 + x_2 & \leq & 1 \end{array}$$

⋆ we have some non-negativity constraints and the main constraints

Geometric solution

⋆ in dimension 2 we can solve the problem by plotting the objective function on the admissible set determined by the constraints!



⋆ observe that in this case the solution is situated at the intersection of the lines

$$5x_1 + 2x_2 = 11 \text{ and } x_1 + 2x_2 = 5.$$

Theoretical observations

⋆ the gradient of $f(x_1, x_2) = x_1 + x_2$ is $(1, 1)$: it is constant and never zero!
⋆ the set K determined by the linear constraints is convex
⋆ the minimum or maximum cannot be attained in the interior of K, since $\nabla f(x) \neq 0$!
⋆ the optimal value is on the boundary of K. Moreover there exists a vertex of the polygon where it can be found! Why?

- start at a point x_0 inside K go against the gradient till you meet an edge
- if the function is constant along an edge then the gradient of the function and the constraint are collinear at that point: Kuhn-Tucker Theorem says that we reached the solution!
- otherwise, follow the direction where the function decreases till reaching a vertex. Then go to the next edge and repeat the previous reasoning.
- the process will finish: finite number of edges!

⋆ same reasoning can be applied in higher dimensions: follow the anti-gradient direction till it is collinear to the gradient of the constraint or no further decrease is possible along further facets!

$\star$ **The Standard Maximum Problem:** Maximize $\mathbf{c}^t\mathbf{x} = c_1x_1 + ... + c_nx_n$ subject to the constraints

$$\begin{aligned} a_{11}x_1 + ... + a_{1n}x_n &\leq b_1 \\ &\vdots \\ a_{m1}x_1 + ... + a_{mn}x_n &\leq b_m \end{aligned} \qquad \text{or } A\mathbf{x} \leq \mathbf{b}$$

and $x_1 \geq 0, x_2 \geq 0, ..., x_n \geq 0$ or $\mathbf{x} \geq 0$

$\star$ **The Standard Minimum Problem:** Minimize $\mathbf{y}^t\mathbf{b} = y_1b_1 + ... + y_mb_m$ subject to the constraints

$$\begin{aligned} a_{11}y_1 + ... + a_{1m}y_m &\geq c_1 \\ &\vdots \\ a_{1n}y_1 + ... + a_{mn}y_m &\geq c_n \end{aligned} \qquad \text{or } y^TA \geq \mathbf{c}^T$$

and $y_1 \geq 0, y_2 \geq 0, ..., y_m \geq 0$ or $\mathbf{y} \geq 0$

$\star$ There are I production sites $P_1, ..., P_I$ which supply a product and J markets $M_1, ..., M_J$ to which the product is shipped.
$\star$ the site P_i contains s_i products and the market M_j must recieve r_j products.
$\star$ the cost of transportation from P_i to M_j is b_{ij}
$\star$ the objective is to minimize the transportation cost while meeting the market requirements!

$\star$ denote by y_{ij} the quantity transported from P_i to M_j. Then the cost is

$$\sum_{i=1}^{I} \sum_{j=1}^{J} y_{ij} b_{ij}$$

$\star$ the constraints are

$$\sum_{j=1}^{J} y_{ij} \leq s_i \text{ and } \sum_{i=1}^{I} y_{ij} \geq r_j.$$

⋆ There are I persons available for J jobs. The "value" of person i working 1 day at job j is a_{ij}.

⋆ **Objective:** Maximize the total "value"

⋆ the variables are x_{ij}: the proportion of person i's time spent on job j

⋆ the constraints are $x_{ij} \geq 0$

$$\sum_{j=1}^{J} x_{ij} \leq 1, i = 1, ..., I \text{ and } \sum_{i=1}^{I} x_{ij} \leq 1,\ j = 1, ..., J \leq 1$$

- can't spend a negative amount of time at a job
- a person can't spend more than 100% of its time
- no more than one person working on a job

Some Terminology

⋆ a point is said to be feasible if it verifies all the constraints
⋆ the set of feasible points is the constraint set
⋆ a linear programming problem is feasible if the constraint set is non-empty. If this is not the case then the problem is infeasible

⋆ every problem involving the minimization of a linear function under linear constraints can be put into standard form

- you can change a "$\geq$" inequality into "$\leq$" by changing the signs of the coefficients
- if a variable x_i has no sign restriction, write it as the difference of two new positive variables $x_i = u_i - v_i,\ u_i, v_i \geq 0$

⋆ it is possible to pass from inequality constraints to equality constraints (and the other way around)

- $Ax = b$ is equivalent to $Ax \leq b$ and $Ax \geq b$
- If $Ax \leq b$ then add some slack variables $\mathbf{u} \geq 0$ such that $Ax + u = b$

Definition 7

The dual of the standard maximum problem

$$\begin{cases} \max \mathbf{c}^T\mathbf{x} \\ \text{s.t. } A\mathbf{x} \le \mathbf{b} \text{ and } \mathbf{x} \ge 0 \end{cases}$$

is the standard minimum problem

$$\begin{cases} \min \mathbf{y}^T\mathbf{b} \\ \text{s.t. } \mathbf{y}^T A \ge \mathbf{c}^T \text{ and } \mathbf{y} \ge 0 \end{cases}.$$

Example

⋆ consider the problem

$$\begin{array}{rl} \text{maximize} & x_1 + x_2 \\ \text{such that} & \mathbf{x} \geq 0 \\ & \begin{array}{rcl} x_1 + 2x_2 & \leq & 5 \\ 5x_1 + 2x_2 & \leq & 11 \\ -2x_1 + x_2 & \leq & 1 \end{array} \end{array}$$

⋆ the dual problem is

$$\begin{array}{rl} \text{minimize} & 5y_1 + 11y_2 + y_3 \\ \text{such that} & \mathbf{y} \geq 0 \\ & \begin{array}{rcl} y_1 + 5y_2 - 2y_3 & \geq & 1 \\ 2y_1 + 2y_2 + y_3 & \geq & 1 \end{array} \end{array}$$

Relation between dual problems

Theorem 8

If $\mathbf{x}$ is feasible for the standard maximum problem and $\mathbf{y}$ is feasible for the dual problem then

$$\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{b}.$$

⋆ The proof is straightforward:

$$\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T A\mathbf{x} \leq \mathbf{y}^T\mathbf{b}.$$

⋆ important consequences:

- if the standard maximum problem and its dual are both feasible, they are bounded feasible: the optimal values are finite!
- If there exist feasible $\mathbf{x}^*$ and $\mathbf{y}^*$ for the standard maximum problem and its dual such that $\mathbf{c}^T\mathbf{x}^* = \mathbf{y}^{*T}\mathbf{b}$ then both are optimal for their respective problems!

Theorem 9 (Duality)

If a standard linear programming problem is bounded feasible then so is its dual, their optimal values are equal and there exist optimal solutions for both problems.

Solve LP problems numerically

⋆ the simplex algorithm: travel along vertices of the set defined by the constraints until no decrease is possible

⋆ work with the matrix A and with vectors $\mathbf{b}$ and $\mathbf{c}$ and modify them using pivot rules: similar to the ones used when solving linear systems

⋆ exploit the connection between the standard formulation and its dual

⋆ things get more complicated when we restrict the variables to be integers. This gives rise to integer programming!

⋆ algorithms solving the main types of LP problems are implemented in various Python packages: `scipy.optimize.linprog`, `pulp`.

The simplex algorithm

⋆ bring the problem to the case of equality constraints using slack variables

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i \iff \sum_{j=1}^{n} a_{ij}x_j + s_i = b_i, s_i \ge 0$$

⋆ any free variable $x_j \in \mathbb{R}$ should be replaced with $u_j - v_j$ with $u_j, v_j \ge 0$

⋆ now we can solve

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \ge 0 \end{aligned}$$

⋆ start from the origin $\mathbf{x} = 0$ and go through the vertices of the polytype $A\mathbf{x} = \mathbf{b}$

⋆ at each step perform an operation similar to the Gauss elimination

⋆ Possible issues: cycling, numerical instabilities.

Practical Example 1

⋆ the first example of a standard maximum problem

$$\max(x_1 + x_2)$$

such that $x_1 \geq 0,\ x_2 \geq 0$ and

$$\begin{array}{rcc} x_1 + 2x_2 & \leq & 5 \\ 5x_1 + 2x_2 & \leq & 11 \\ -2x_1 + x_2 & \leq & 1 \end{array}$$

⋆ we saw geometrically that the solution should be the intersection of $x_1 + 2x_2 = 5$ and $5x_1 + 2x_2 = 11$

```
scipy.optimize.linprog(c, A_ub=None, b_ub=None, A_eq=None,
            b_eq=None,bounds=None, method='simplex',
            callback=None, options=None)
```

Practical Example 2

⋆ An optimal assignment problem: n

	Job 1	Job 2	Job 3
Person 1	100€	120€	80€
Person 2	150€	110€	120€
Person 3	90€	80€	110€

⋆ assign Person i to Job j in order to minimize the total cost!

⋆ we can model the situation as an LP problem with 9 variables: $x_{ij} = 1$ if and only if Person i has job j, $1 \leq i, j \leq 3$

⋆ the constraints are as follows:

- $\sum_{i=1}^{3} x_{ij} = 1$: exactly one Person for Job j
- $\sum_{j=1}^{3} x_{ij} = 1$: exactly one Job for Person i

⋆ we should also impose that $x_i \in \{0, 1\}$: no fractional jobs, but we'll neglect this condition and just suppose $x_i \geq 0$.

⋆ the cost is just

$$\sum_{1 \leq i,j \leq 3} c_{ij} x_{ij}$$

Find the LP parameters

$\star$ let's look at the matrix of the problem: 9 variables and 6 constraints!

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$\star$ the matrix c_{ij} is given by the table shown previously: the cost of every person per function $\star$ the vector b is equal to 1 on every component

$\star$ the solution is made of zeros and ones, without imposing this...
$\star$ this phenomenon always happens: if A is a totally unimodular matrix and b is made of integers then $Ax \leq b$ has all its vertices at points with integer coordinates

A matrix is totally unimodular if every square submatrix has determinant in the set $\{0, 1, -1\}$.

⋆ solving a Sudoku with LP

					3		8	5
		1		2				
			5		7			
		4				1		
	9							
5							7	3
		2		1				
				4				9

⋆ Remember the rules: $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ should be found on every line, column and 3×3 square

⋆ in order to make this solvable via LP a different formulation should be used!

⋆ classical idea: use binary variables

Sudoku in Binary variables

⋆ how to represent a Sudoku puzzle using 0s and 1s?

⋆ build a 3D array $X = (x_{ijk})$ of size $9 \times 9 \times 9$ such that

$x_{ijk} = 1$ if and only if on position (i, j) we have the digit k; else $x_{ijk} = 0$

⋆ what are the constraints in this new formulation?

- $x_{ijk} \in \{0, 1\}$: again to be relaxed to $x_{ijk} \geq 0$ - 729 constraints
- fixing i, j: $\sum_{k=1}^{9} x_{ijk} = 1$ - one number per cell - 81 constraints
- fixing i, k: $\sum_{j=1}^{9} x_{ijk} = 1$ - k appears exactly once on line i - 81 constraints
- fixing j, k: $\sum_{i=1}^{9} x_{ijk} = 1$ - k appears exactly once on column j - 81 constraints
- small 3×3 squares condition: for $u, v \in \{0, 3, 6\}$

$$\sum_{i=1}^{3} \sum_{j=1}^{3} x_{i+u, j+v, k} = 1, \ k = 1, ..., 9 \text{ - 81 constraints}$$

- the initial given information for the puzzle may be written in the form $s_{ij} = k$ for some i, j, k. This gives the constraints $x_{i,j,s_{ij}} = 1$.

⋆ we are interested in finding a feasible solution: no objective function is needed!

Solving the Sudoku

⋆ a feasible solution can be found using the simplex algorithm
⋆ sometimes we may get non-integer results: apparently, the constraint matrix is not always a Totally Unimodular matrix
⋆ there are LP algorithms which will return integer solutions: integer programming

⋆ before solving we should check that the constraint matrix should be of maximal rank: eliminate redundant constraints
⋆ we could also eliminate fixed variables: the data $s_{ij} = k$ should eliminate all unknowns with first index i, second index j or third index k!

⋆ if the solution is unique: the algorithm will find it
⋆ if the solution is not unique: the algorithm will find one of the solutions. We may repeat with the constraint that the solution should be different than the previous one, until no other solutions are found!
⋆ check out the PuLP Python library: an example of Sudoku solver is given!

Conclusions on LP

- minimize/maximize linear functions under linear constraints
- many practical applications from an industrial point of view!
- there exist optimizers which are vertices of the constraint set
- simplex algorithm: travel along vertices decreasing the objective function
- computational complexity: worst case is exponential: Klee-Minty cube
- polynomial-time average case complexity: most of the LP problems will be solved very fast!

Conclusion of the course

⋆ numerical optimization (unconstrained case):

- derivatives-free algorithms: no-regularity needed, slow convergence
- gradient descent algorithms: linear convergence, sensitive to the condition number
- Newton, quasi-Newton: super-linear convergence in certain cases
- when dealing with large problems use L-BFGS
- Conjugate Gradient: solve linear systems, better than GD
- Gauss-Newton: useful when minimizing a non-linear least squares function

⋆ constrained case

- for simple constraints: use the projected gradient algorithm
- general smooth constraints: use the tangential part of the gradient and come back to the constraint set using the Newton method
- other options available: SQP, etc...
- Linear Programming: use specific techniques: the simplex algorithm $\longrightarrow$ to be continued next year in the course dealing with Convex Optimization!

Conclusion of the course

- know your options when looking at an optimization problem: choose the right algorithm depending on: the size of the problem, the number of variables, the regularity, the conditioning, etc.
- learn how to use existing solutions: `scipy.optimize` is a good starting point
- know how to code your own optimization algorithm if necessary: use gradients when possible, limit the number of function evaluations, choose a good stopping criterion, limit the number of iterations, etc.