Programming, numerics and optimization

Lecture C-4: Constrained optimization

Łukasz Jankowski
ljank@ippt.pan.pl

Institute of Fundamental Technological Research
Room 4.32, Phone +22.8261281 ext. 428

May 18, 2021[1]

[1] Current version is available at http://info.ippt.pan.pl/~ljank.

Outline

Outline

1. Basics
   - Optimization problem
   - Active and inactive constraints
   - Local and global minima
   - Constrained vs. unconstrained optimization

Optimization problem

Optimization problem = objective function + domain

- *Objective function* $f : \mathbf{D} \to \mathbb{R}$ is the function to be minimized
- *Domain* $\mathbf{D}$ of the objective function arguments
  - *Unconstrained*, $\mathbf{D} = \mathbb{R}^n$
  - *Constrained* with
    - equality constraints, $\mathbf{D} = \{\mathbf{x} \in \mathbb{R}^n | g_i(\mathbf{x}) = 0, i = 1, \ldots, m_1\}$
    - or inequality constraints, $\mathbf{D} = \{\mathbf{x} \in \mathbb{R}^n | g_i(\mathbf{x}) \geq 0, i = m_1 + 1, \ldots, m_1 + m_2\}$

  where g_i are called the *constraint functions*.

Optimization problem (minimization)

Find $\hat{\mathbf{x}} \in \mathbf{D}$ such that

$$\forall_{\mathbf{x} \in \mathbf{D}} \, f(\mathbf{x}) \geq f(\hat{\mathbf{x}})$$

Active and inactive constraints

Each point $\mathbf{x}$ in the domain, $\mathbf{x} \in \mathbf{D}$, is called a *feasible point*.

An inequality constraint at a feasible point can be either

active $g_i(\mathbf{x}) = 0$ or

inactive $g_i(\mathbf{x}) > 0$.

The equality constraints are always active at all feasible points.

Local and global minima

Local minimum

$\hat{\mathbf{x}} \in \mathbf{D}$ is called a *local minimum* of f, iff there exists a neighborhood $\mathbf{\Omega} \subset \mathbb{R}^n$ of $\hat{\mathbf{x}}$ (that is, an open set containing $\hat{\mathbf{x}}$), such that $\hat{\mathbf{x}}$ minimizes f within $\mathbf{\Omega} \cap \mathbf{D}$:

$$\forall_{\mathbf{x} \in \mathbf{\Omega} \cap \mathbf{D}} \; f(\mathbf{x}) \geq f(\hat{\mathbf{x}})$$

Global minimum

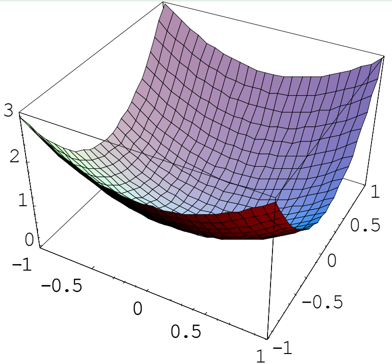
$\hat{\mathbf{x}} \in \mathbf{D}$ is called a *global minimum* of f, iff it minimizes f within the whole domain $\mathbf{D}$:

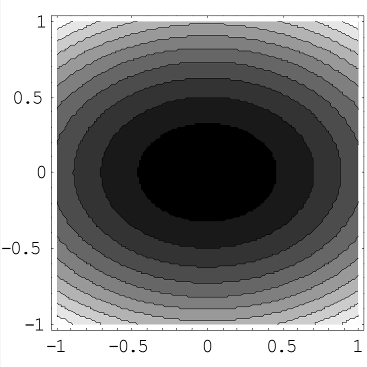
$$\forall_{\mathbf{x} \in \mathbf{D}} \; f(\mathbf{x}) \geq f(\hat{\mathbf{x}})$$

Constrained vs. unconstrained optimization

Unconstrained optimization: an easy case

$$f(x, y) = x^2 + 2y^2 \qquad \text{(one global minimum)}$$



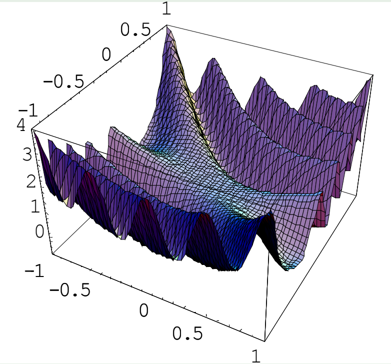


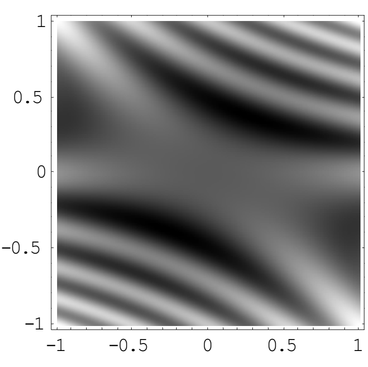
Constrained vs. unconstrained optimization

Unconstrained optimization: a hard case

$$f(x, y) = x^2 + 2y^2 + \cos 4\pi(x + y)y$$

(infinitely many local minima, two global)

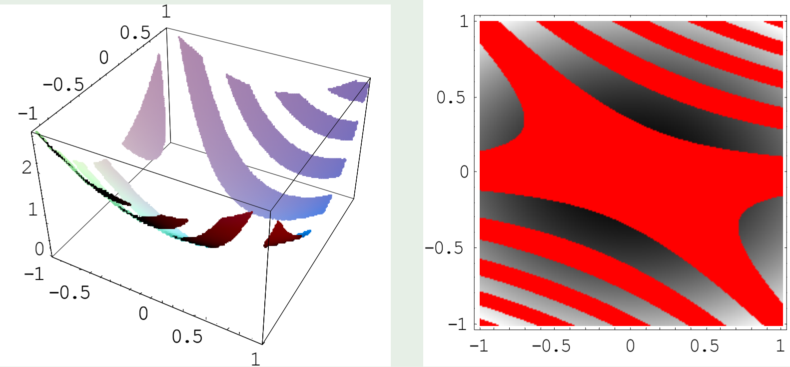




Constrained vs. unconstrained optimization

Constraints may turn an easy case into a hard case

$$f(x,y) = x^2 + 2y^2, \qquad \cos 4\pi(x+y)y \le 0$$

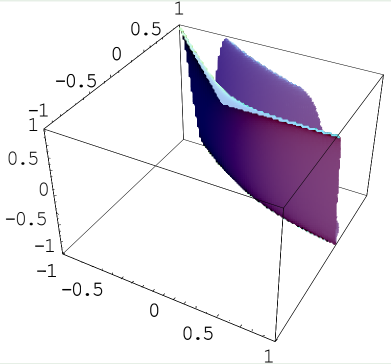


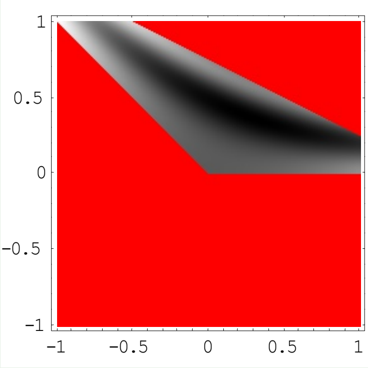
Constrained vs. unconstrained optimization

Constraints may also turn a hard case into an easy case

$$f(x, y) = x^2 + 2y^2 + \cos 4\pi(x + y)y$$

$$x + y \geq 0,\ \ y \geq 0,\ \ 2x + 4y \leq 3$$





Outline

2. Handling constraints
   - Optimality criteria in unconstrained optimization
   - Lagrangian and KKT conditions
   - Penalty functions

Optimality criteria in unconstrained optimization

If the objective function f is smooth enough, the necessary and sufficient first- and second order optimality criteria in unconstrained optimization can be stated in a simple form:

necessary 1st order: If $\hat{\mathbf{x}}$ is a local minimum of f, then $\nabla f(\hat{\mathbf{x}}) = \mathbf{0}$ (that is, a minimum must be a *stationary point*).

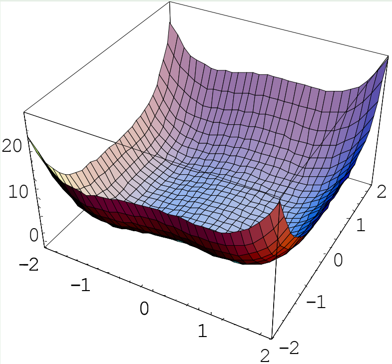
sufficient 2nd order: If $\nabla f(\hat{\mathbf{x}}) = \mathbf{0}$ and $\nabla^2 f(\hat{\mathbf{x}})$ is positive definite, then $\hat{\mathbf{x}}$ is a (strict) local minimum of f.

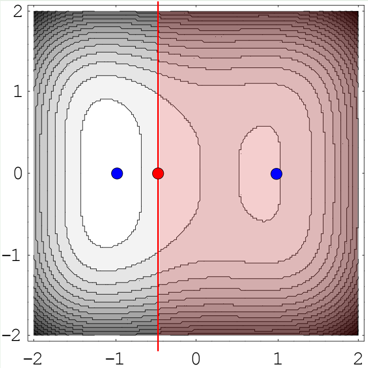
However, in constrained optimization only the sufficient criterion holds. The necessary criterion is not valid: in a local minimum the gradient may be non-vanishing.

Optimality criteria in unconstrained optimization

The necessary criterion may fail in the constrained case

$$f(x, y) = x + y^4 + (x^2 - 1)^2, \qquad x \geq -\frac{1}{2}$$

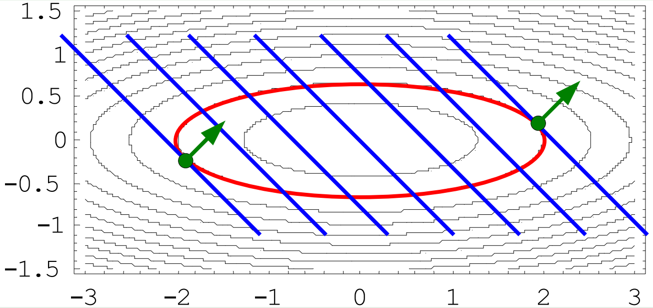




Equality constraints

An equality constrained problem

$$f(x, y) = x + y, \qquad g(x, y) = x^2 + 9y^2 - 4 = 0$$

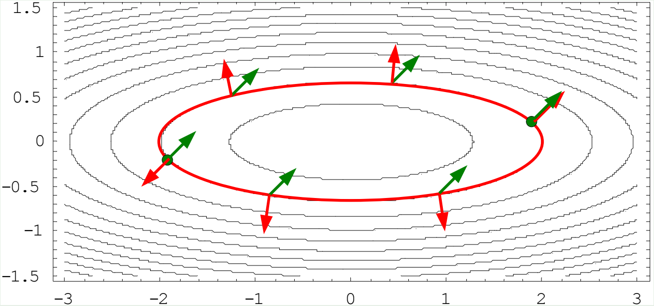


At the minimum the isolines of the objective function are tangent to the constraint.

Equality constraints

An equality constrained problem

$$f(x, y) = x + y, \qquad g(x, y) = x^2 + 9y^2 - 4 = 0$$



At the minimum the isolines of the objective function are tangent to the constraint. In other terms, the gradients of the objective function and of the constraint are co-linear.

Equality constraints

Consider an optimization problem with one regular[2] equality constraint $g(\mathbf{x}) = 0$. Let $\mathbf{x}$ be feasible. Any direction $\mathbf{d}$ that

- upholds the constraint $\nabla g(\mathbf{x})^\mathsf{T}\mathbf{d} = 0$ and
- is a descent direction $\nabla f(\mathbf{x})^\mathsf{T}\mathbf{d} < 0$.

can be used to further decrease the objective function. Hence a necessary condition for optimality is: there is no direction $\mathbf{d}$ satisfying both of the above. And it is possible only when $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are co-linear, that is when

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x})$$

for some λ. This amounts to the condition $\nabla_\mathbf{x} L(\mathbf{x}, \lambda) = \mathbf{0}$, where $L(\mathbf{x}, \lambda)$ is the Lagrangian,

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x}).$$

[2] Regular, that is with non-vanishing and continuous gradient ∇g.

The method of Lagrange multipliers

The method of Lagrange multipliers

Assume $\hat{\mathbf{x}}$ is a local minimum of $f(\mathbf{x})$, subject to

$$g_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, m_1,$$

and the gradients of g_i are all continuous and non-vanishing. Then there exists a vector $\boldsymbol{\lambda}$ of Lagrange multipliers such that

$$\nabla_{\mathbf{x}} L(\hat{\mathbf{x}}, \boldsymbol{\lambda}) = \mathbf{0},$$

where $L(\mathbf{x}, \boldsymbol{\lambda})$ is the Lagrangian:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{i=1}^{m_1} \lambda_i g_i(\mathbf{x}).$$

Inequality constraints

Consider an optimization problem with one regular inequality constraint $g(\mathbf{x}) \geq 0$. Let $\mathbf{x}$ be feasible. Any direction $\mathbf{d}$ that

- upholds the constraint $g(\mathbf{x}) + \nabla g(\mathbf{x})^T \mathbf{d} \geq 0$ and
- is a descent direction $\nabla f(\mathbf{x})^{\mathsf{T}} \mathbf{d} < 0$.

can be used to further decrease the objective function.

These conditions amount to

inactive constraint $(g(\mathbf{x}) > 0)$ a single requirement $\nabla f(\mathbf{x})^{\mathsf{T}} \mathbf{d} < 0$ (since the constraint holds for all $\mathbf{d}$ short enough).

active constraint $(g(\mathbf{x}) = 0)$ both conditions simplify to $\nabla f(\mathbf{x})^{\mathsf{T}} \mathbf{d} < 0$ and $\nabla g(\mathbf{x})^T \mathbf{d} \geq 0$.

Inequality constraints

The direction $\mathbf{d}$ is a feasible descent direction, if

case $g(\mathbf{x}) > 0$ $\nabla f(\mathbf{x})^\mathsf{T}\mathbf{d} < 0$

case $g(\mathbf{x}) = 0$ $\nabla f(\mathbf{x})^\mathsf{T}\mathbf{d} < 0$ and $\nabla g(\mathbf{x})^T\mathbf{d} \geq 0$.

A necessary condition for optimality (that is, for no feasible descent direction) is hence:

case $g(\mathbf{x}) > 0$ $\nabla f(\mathbf{x}) = 0$

case $g(\mathbf{x}) = 0$ $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are co-linear and point in the same direction: $\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x})$ for some $\lambda > 0$.

The necessary condition for optimality can be expressed in a form common to both cases as:

$$\nabla_\mathbf{x} L(\mathbf{x}, \lambda) = \mathbf{0} \qquad \text{and} \qquad \lambda g(\mathbf{x}) = 0$$

for some $\lambda > 0$. $\lambda g(\mathbf{x}) = 0$ implies that λ can be positive only when the constraint is active.

Karush-Kuhn-Tucker (KKT) conditions

In the general case, minimize f subject to

$$\begin{aligned} g_i(\mathbf{x}) &\geq 0 & i &= 1, 2, \ldots, m_1 \\ g_i(\mathbf{x}) &= 0 & i &= m_1 + 1, \ldots, m_1 + m_2, \end{aligned}$$

and provided the gradients of the constraints are continuous and regular enough,

> *In fact, many formalizations of this "regularity" requirement are possible, the most common is called LICQ (Linear Independence Constraint Qualification): the gradients of the active constraints are linearly independent.*

then the necessary first-order conditions for optimality can be stated in the form of the Karush-Kuhn-Tucker (KKT) conditions.

Karush-Kuhn-Tucker (KKT) conditions

Karush-Kuhn-Tucker (KKT) conditions

Assume $\hat{\mathbf{x}}$ is a local minimum of $f(\mathbf{x})$, subject to

$$
\begin{aligned}
g_i(\mathbf{x}) &\geq 0 \qquad && i = 1, 2, \ldots, m_1 \\
g_i(\mathbf{x}) &= 0 \qquad && i = m_1 + 1, \ldots, m_1 + m_2,
\end{aligned}
$$

the gradients of g_i are continuous and LICQ holds at $\hat{\mathbf{x}}$. Then there exists a unique vector $\boldsymbol{\lambda}$ of Lagrange multipliers such that, for $i = 1, 2, \ldots, m_1$, $\lambda_i \geq 0$ and

$$\nabla_{\mathbf{x}} L(\hat{\mathbf{x}}, \boldsymbol{\lambda}) = \mathbf{0} \qquad \text{and} \qquad \lambda_i g_i(\hat{\mathbf{x}}) = 0,$$

where $L(\mathbf{x}, \boldsymbol{\lambda})$ is the Lagrangian:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{i=1}^{m_1+m_2} \lambda_i g_i(\mathbf{x}).$$

Hard and soft constraints

All constraints can be categorized into two general classes:

1. A constraint is a *hard constraint*, if it cannot be broken. For example, an objective function involving a logarithm or a square root is undefined for negative arguments. All points generated by an optimization algorithm must be feasible with respect to the hard constraints.
2. A constraint is called a *soft constraint*, if it can be broken during the search. For example, a required upper bound on the total cost can be broken during the search, provided that the solution obeys it. Some optimization algorithms may (temporarily) generate infeasible points with respect to the soft constraints to speed-up the optimization process.

Penalty functions

Constraints can be effectively dropped by adding to the objective function a *penalty function* to penalize the points that approach or break the constraints:

- Exterior quadratic penalty function (soft constraints only):

$$f_{\mathrm{p}}(\mathbf{x}, \alpha) = f(\mathbf{x}) + \frac{1}{2\alpha} \sum_{i=1}^{m_1} g_i^2(\mathbf{x}) + \frac{1}{2\alpha} \sum_{i=m_1+1}^{m_2} \left[\min(0, g_i(\mathbf{x}))\right]^2 ,$$

- Interior log and inverse barriers (inequality constraints only):

$$\begin{aligned} f_{\mathrm{p}}(\mathbf{x}, \alpha) &= f(\mathbf{x}) - \alpha \textstyle\sum_{i=1}^{m_2} \log g_i(\mathbf{x}), \\ f_{\mathrm{p}}(\mathbf{x}, \alpha) &= f(\mathbf{x}) + \alpha \textstyle\sum_{i=1}^{m_2} \left[g_i(\mathbf{x})\right]^{-1} . \end{aligned}$$

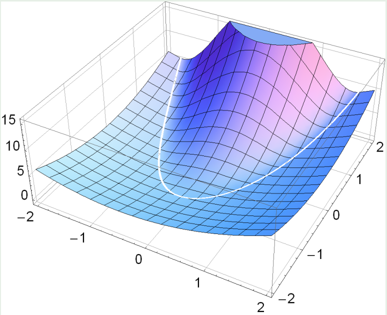
The penalty function can be added also to the Lagrangian instead of the objective function. This yields the method of augmented Lagrangian.

Penalty functions

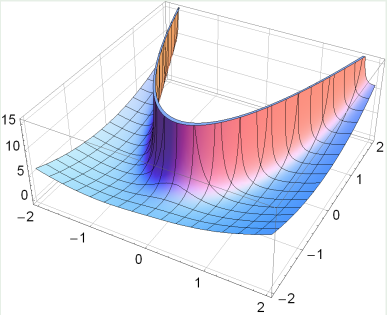
Minimization problem

$$f(x, y) = x(x+1) + y^2 \qquad x^2 - y - 1 \geq 0$$

Quadratic penalty function



Inverse barrier



Penalty functions

Penalty functions:

- Transform constrained problems into unconstrained, which are easier to handle.
- In general, the minimum of the augmented function is found, which only approximates the minimum of the objective function. The optimization usually has to be repeated with gradually steeper barrier slope, that is with α decreasing to zero.
- Increase the ill-conditioning of the problem (the augmented Lagrangian method performs often better with respect to conditioning).

Outline

3. Types of problems

Types of problems

Linear programming the function and the constraints are linear

Quadratic programming the function is quadratic and the constraints linear

Nonlinear programming the function or at least one of the constraints is non-linear

Convex programming the function and the domain are convex (linear equality constraints, concave inequality constraints).

Bound-constrained optimization upper or lower bounds on variables are the only constraints

etc.

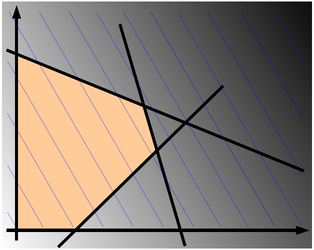
Outline

4. Linear programming
   - Formulation of the problem
   - Methods
   - KKT conditions
   - Simplex method
   - Interior point methods

Linear programming — the problem

Linear programming (LP)

$$\begin{array}{rl} \text{Minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}. \end{array}$$



- The objective function and the constraints are linear.
- The domain is defined by a set of (standarized) linear constraints. All general linear constraints ($\mathbf{A}\mathbf{x} \geq \mathbf{b}$) can be converted to the standard form by introducing slack variables.
- In geometric terms the domain is a closed convex polytope (an intersection of hyper-planes). In practical cases it is bounded and non-empty.

Linear programming — constraint standarization

Minimize $\mathbf{c}^T\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$.

The following more general linear constraints

$$\mathbf{A}\mathbf{x} \geq \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0}$$

can be converted to the standard form by introducing slack variables $\mathbf{w}$,

$$\mathbf{A}\mathbf{x} - \mathbf{w} = \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0},\ \mathbf{w} \geq \mathbf{0}.$$

Linear programming — constraint standarization

Minimize $\mathbf{c}^\mathsf{T}\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$.

Another form of the constraints,

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \ \text{(without the bounds)}$$

can be converted to the standard form by splitting all x_i into their nonnegative and nonpositive parts, $x_i = x_i^+ - x_i^-$, where

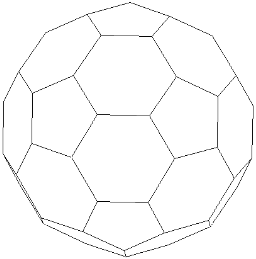
$$x_i^+ = \max\{0, x_i\} \geq 0, \qquad x_i^- = -\min\{0, x_i\} \geq 0.$$

In the new variables the problem takes the standard form

$$\text{Minimize} \begin{bmatrix} \mathbf{c} & -\mathbf{c} \end{bmatrix} \begin{bmatrix} \mathbf{x}^+ \\ \mathbf{x}^- \end{bmatrix}, \text{ subject to} \begin{bmatrix} \mathbf{A} & -\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x}^+ \\ \mathbf{x}^- \end{bmatrix} = \mathbf{b},$$
$$\mathbf{x}^+ \geq \mathbf{0}, \ \mathbf{x}^- \geq \mathbf{0}.$$

Linear programming — methods

The objective function and the constraints are linear, hence the minimum is attained *in the vertices* of the boundary of the domain (provided it is bounded and non-empty).



LP algorithms fall into two classes:

1. Some start and remain on the boundary of the domain polytope, moving through the vertices only and choosing in each successive step an edge leading to a “better” vertex (simplex method).
2. Some use also interior points to speed-up the computations (interior point methods).

Linear programming — KKT conditions

Minimize $\mathbf{c}^\mathsf{T}\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$

The Lagrangian: $L(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}) = \mathbf{c}^\mathsf{T}\mathbf{x} - \boldsymbol{\pi}^\mathsf{T}(\mathbf{A}\mathbf{x} - \mathbf{b}) - \boldsymbol{\kappa}^\mathsf{T}\mathbf{x}$.

The KKT conditions are both *necessary* and *sufficient*: $\hat{\mathbf{x}}$ is a global minimum, if and only if there exist vectors $\boldsymbol{\kappa}$ and $\boldsymbol{\pi}$ such that

$$\begin{aligned} \mathbf{A}^\mathsf{T}\boldsymbol{\pi} + \boldsymbol{\kappa} &= \mathbf{c}, \\ \boldsymbol{\kappa} &\geq \mathbf{0}, \\ \hat{x}_i \kappa_i &= \mathbf{0} \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

and $\hat{\mathbf{x}}$ is feasible, that is

$$\begin{aligned} \mathbf{A}\hat{\mathbf{x}} &= \mathbf{b}, \\ \hat{\mathbf{x}} &\geq \mathbf{0}. \end{aligned}$$

Simplex method

> *Dantzig's[3] development of the simplex method in the late 1940s marks the start of the modern era in optimization. This method made it possible for economists to formulate large models and analyze them in a systematic and efficient way. Dantzig's discovery coincided with the development of the first digital computers, and the simplex method became one of the earliest important applications of this new and revolutionary technology.*

Jorge Nocedal, Stephen Wright
Numerical Optimization

[3] George B. Dantzig (1914–2005), a "founding father" of linear programming and the simplex method.

Simplex method

The simplex method starts in a vertex of the domain and moves through the vertices choosing in each successive step an edge leading to a "better" vertex (e.g. a vertex with a lower or equal value of the objective function).

Assume the matrix $\mathbf{A}$ is $m \times n$ and full row rank. A feasible point $\mathbf{x}$ is a vertex, if

- It has at most m non-zero components x_i.
- There exists an m-element set $\mathcal{I}_{\mathbf{x}} \subset \{1, 2, \ldots, n\}$ containing indices i of all non-zero components x_i such that the matrix $\mathbf{A}_{\mathcal{I}_{\mathbf{x}}}$ composed of the corresponding columns of $\mathbf{A}$ is nonsingular.

Moving from a vertex to an adjacent vertex corresponds to replacing one index in the index set $\mathcal{I}_{\mathbf{x}}$ and recomputing the components of $\mathbf{x}$.

Simplex method

The simplex method has some non-trivial points:

- Finding the initial vertex can be difficult.
- There are several alternative rules to choose the best from the adjacent vertices.

Moreover, in some large practical problems cycling may occur: several (zero-length) degenerate steps can lead back to an already visited vertex. An anticycling strategy is then necessary.

It turned out that the simplex method has an important deficiency: although in almost all practical problems it is very quick, the general time complexity is exponential. There are (rather artificial) problems, in which the method visits all vertices before reaching the optimum.

Interior point methods

The in general exponential time complexity of the simplex method has motivated the development of interior point methods with polynomial complexity:

- The *simplex method* takes a lot of easy-to-compute steps around the boundary of the domain, while
- the *interior point methods* use a smaller number of expensive steps through interior points of the domain and approach the optimum point of the boundary only in the limit.

Interior point methods

The feasibility and KKT optimality conditions for the LP problem of minimizing $\mathbf{c}^{\mathsf{T}}\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ are

$$\begin{aligned} \mathbf{A}^{\mathsf{T}}\boldsymbol{\pi} + \boldsymbol{\kappa} &= \mathbf{c}, \\ \mathbf{A}\mathbf{x} &= \mathbf{b}, \\ x_i \kappa_i &= 0 \qquad \text{for } i = 1, 2, \ldots, m \end{aligned}$$

and

$$\begin{aligned} \mathbf{x} &\geq \mathbf{0}, \\ \boldsymbol{\kappa} &\geq \mathbf{0}. \end{aligned}$$

The most popular class of interior point methods (*primal-dual methods*) solves the equality conditions using nonlinear Newton's method and biasing the search direction to strictly obey the inequality conditions.

Primal-dual methods

The equality conditions can be stated as

$$F(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}) = \begin{bmatrix} \mathbf{A}^{\mathsf{T}}\boldsymbol{\pi} + \boldsymbol{\kappa} - \mathbf{c} \\ \mathbf{A}\mathbf{x} - \mathbf{b} \\ \mathbf{X}\mathbf{K}\mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

where

$$\begin{aligned} \mathbf{X} &= \operatorname{diag}\{x_1, \ldots, x_n\}, \\ \mathbf{K} &= \operatorname{diag}\{\kappa_1, \ldots, \kappa_n\}, \\ \mathbf{e} &= [1, \ldots, 1]^{\mathsf{T}}. \end{aligned}$$

Due to the term $\mathbf{XK}$ these equations are mildly nonlinear. The primal-dual methods solve it using nonlinear Newton's method.

Primal-dual methods

Nonlinear Newton's method linearizes the equation,

$$\begin{aligned}\mathbf{0} &= F(\mathbf{x}+\Delta\mathbf{x}, \boldsymbol{\pi}+\Delta\boldsymbol{\pi}, \boldsymbol{\kappa}+\Delta\boldsymbol{\kappa}) \\ &\approx F(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}) + J(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}) \begin{bmatrix} \Delta\mathbf{x} & \Delta\boldsymbol{\pi} & \Delta\boldsymbol{\kappa} \end{bmatrix}^{\mathsf{T}},\end{aligned}$$

that is

$$J(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}) \begin{bmatrix} \Delta\mathbf{x} & \Delta\boldsymbol{\pi} & \Delta\boldsymbol{\kappa} \end{bmatrix}^{\mathsf{T}} = -F(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}),$$

and solves it to find the search direction. For primal-dual feasible[4] $\mathbf{x}$, $\boldsymbol{\pi}$ and $\boldsymbol{\kappa}$ it reduces to solving

$$\begin{bmatrix} \mathbf{0} & \mathbf{A}^{\mathsf{T}} & \mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{K} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\pi} \\ \Delta\boldsymbol{\kappa} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{XKe} \end{bmatrix}.$$

[4] That is, for $\mathbf{x}$, $\boldsymbol{\pi}$ and $\boldsymbol{\kappa}$ feasible for both the primal problem ($\mathbf{Ax} = \mathbf{b}$) and the dual problem ($\mathbf{A}^{\mathsf{T}}\boldsymbol{\pi} + \boldsymbol{\kappa} = \mathbf{c}$).

Primal-dual methods — biasing the search direction

The direction computed by solving

$$J(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}) \begin{bmatrix} \Delta\mathbf{x} & \Delta\boldsymbol{\pi} & \Delta\boldsymbol{\kappa} \end{bmatrix}^{\mathsf{T}} = -F(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa})$$

is used as a linear search direction.

As the number of iterations grows, $x_i \kappa_i \to 0$, and usually only a small step can be taken before the bounds $\mathbf{x} \geq \mathbf{0}$ and $\boldsymbol{\kappa} \geq \mathbf{0}$ are violated. The primal-dual methods bias thus the computed direction to keep a proper distance from the bounds.

Primal-dual methods — the central path method

The central path is an arc of feasible points parametrized by $\tau > 0$, which ensures a proper distance from the bounds $\mathbf{x} \geq \mathbf{0}$ and $\boldsymbol{\kappa} \geq \mathbf{0}$:

$$\begin{aligned} \mathbf{A}^\mathsf{T}\boldsymbol{\pi} + \boldsymbol{\kappa} &= \mathbf{c}, \\ \mathbf{A}\hat{\mathbf{x}} &= \mathbf{b}, \\ \hat{x}_i \kappa_i &= \tau, \\ \hat{\mathbf{x}} &\geq \mathbf{0}, \\ \boldsymbol{\kappa} &\geq \mathbf{0}. \end{aligned}$$

As $\tau \to 0$, the central path can converge only to the solution.

The method biases the direction towards a point on the central path defined by a certain $\tau > 0$, instead of the exact solution ($\tau = 0$), in order to equalize the distance from all the bounds and allow for much longer steps in further iterations.

Primal-dual methods — the central path method

The central path method computes the biased direction by solving

$$F(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\kappa}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \sigma\mu\mathbf{e} \end{bmatrix}^\mathsf{T},$$

where

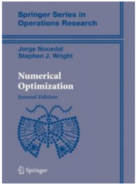
$$\mathbf{e} = [1, \ldots, 1]^\mathsf{T},$$
$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \kappa_i = \frac{\mathbf{x}^\mathsf{T}\boldsymbol{\kappa}}{n}$$

and $\sigma \in [0, 1]$ is a parameter weighting between centering ($\sigma = 1$) and progressing towards the exact solution $\sigma = 0$.

Outline

5. Further reading

Further reading



Jorge Nocedal, Stephen Wright
Numerical Optimization
2nd edition, Springer 2006.

Outline

6 Homework 9

Homework 9 (15 points + 5 extra)

Linear programming

1. Consider the following linear problem:

$$\begin{array}{ll} \text{Maximize} & 2x + 2y + z \\ \text{subject to} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} 5 \\ 4 \\ 4 \end{bmatrix} \\ & x \geq 0, \;\; y \geq 0, \;\; z \geq 0. \end{array}$$

   1. (5 points) The constraints define a 3D polytope. What shape is it? What is the number of its vertices, edges and facets?
   2. (5 points) Solve the problem.

Homework 9 (15 points + 5 extra)

Linear programming

2. An intersection of a 2D plane

$$ax + by + cz = d$$

and a unit cube

$$\Big\{\{x, y, z\} \in \mathbb{R}^3 \big| \, 0 \leq x, y, z \leq 1\Big\}$$

is a (sometimes degenerate) 2D polytope.

1. (5 points) What types of polytopes can be obtained this way (triangle? square? ...?) For each polytope type state the equation of the corresponding plane.
2. (5 points extra) What types of polytopes can be obtained by intersecting a 2D plane with a four dimensional cube?

E-mail the answers to `ljank@ippt.pan.pl`.