Discrete Optimization and Integer Programming.

Classical optimization methods

“Nothing takes place in the world whose meaning is not that of some maximum or minimum.” (L. Euler)

Contents

I. Necessary and Sufficient Conditions for Extrema

- ***Unconstrained Optimization***

$$f(x) \to min, \qquad x \in \mathbb{R}^n$$

Theorem 1. Let $x^* \in \mathbb{R}^{\boldsymbol{n}}$ be a local minimum (maximum) of the function $f(x)$ and this function is differentiable at the point x^*. Then we have the equality

$$\nabla f(x^*) = 0.$$

Theorem 2. Let $x^* \in \mathbb{R}^{\boldsymbol{n}}$ be a local minimum (maximum) of the function $f(x)$ and this function is twice differentiable at the point x^*. Then we have the equality

$$\mathrm{Hesse}(f)(x^*) \geq 0.$$

Theorem 3. Let the function $f(x)$ is twice differentiable at the point $x^* \in \mathbb{R}^{\boldsymbol{n}}$ and we have

$$\nabla f(x^*) = 0, \qquad \mathrm{Hesse}(f)(x^*) \geq 0.$$

Then the point $x^* \in \mathbb{R}^{\boldsymbol{n}}$ is a local minimum of the function $f(x)$ on the set $\mathbb{R}^{\boldsymbol{n}}$.

I. Necessary and Sufficient Conditions for Extrema

- ***Unconstrained Optimization***

$\nabla f(x^*)$	$\mathrm{Hesse}(f)(x^*)$	Condition type	Minors	Eigenvalues	Type of critical point
0	> 0	Sufficient	$\Delta_1 > 0, \Delta_2 > 0, \ldots, \quad \Delta_n > 0$	$\lambda_1 > 0, \ldots, \lambda_n > 0$	Local minimum
0	< 0	Sufficient	$\Delta_1 < 0, \Delta_2 > 0, \ldots,$ $(-1)^n \ \Delta_n > 0$	$\lambda_1 < 0, \ldots, \lambda_n < 0$	Local maximum
0	≥ 0	Necessary	All principal minors ≥ 0	$\lambda_1 \geq 0, \ldots, \lambda_n \geq 0$	Probably local minimum
0	≤ 0	Necessary	All principal minors of even order ≥ 0, all principal minors of odd order are ≤ 0	$\lambda_1 \leq 0, \ldots, \lambda_n \leq 0$	Probably local maximum
0	$= 0$	Necessary	All elements are zero	$\lambda_1 = 0, \ldots, \lambda_n = 0$	Need additional information
0	$\lessgtr 0$	Necessary	---	$\lambda_i \gtrless 0$	Not extremal point

I. Necessary and Sufficient Conditions for Extrema

- ***Equality Constraints. Lagrange Multipliers.***

$$f(x) \to min, \qquad x \in X \subset \mathbb{R}^n,$$

$$X = \{g_j(x) = 0, j = 1, \dots, m\}.$$

Definition 1. The function

$$L(x, \lambda_0, \lambda) := \lambda_0 f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

is called *Lagrange function*, the numbers $\lambda_0, \lambda_1, \dots, \lambda_m$ are called *Lagrange multipliers*. Also we will consider the *regular Lagrange function*

$$L(x, \lambda) \coloneqq L(x, 1, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x).$$

I. Necessary and Sufficient Conditions for Extrema

- ***Equality Constraints. Necessary Conditions.***

Theorem 4. Let the point $x^* \in X$ be a local minimum of the function $f(x)$ on the set $X \subset \mathbb{R}^n$. Then there exists the numbers $\lambda_0^*, \lambda_1^*, \dots, \lambda_m^*$ not equal to zero at the same time such that we have

$$\frac{\partial L(x^*, \lambda_0^*, \lambda^*)}{\partial x_i} = 0, \qquad i = 1, \dots, n;$$

$$g_j(x^*) = 0, \qquad j = 1, \dots, m.$$

Moreover, if $\nabla g_1(x^*), \dots, \nabla g_m(x^*)$ are linear independent then $\lambda_0^* \neq 0$.

Theorem 5. Let the point $x^* \in X$ be a regular (i. e. $\lambda_0^* \neq 0$) local minimum of the function $f(x)$ on the set $X \subset \mathbb{R}^n$ and λ^* be the vector from the previous theorem. Then we have

$$d^2 L(x^*, \lambda^*) \geq 0$$

for all $dx \in T_{x^*}\mathbb{R}^n$ such that

$$dg_j(x^*) = 0, j = 1, \dots, m.$$

I. Necessary and Sufficient Conditions for Extrema

- ***Equality Constraints. Sufficient conditions.***

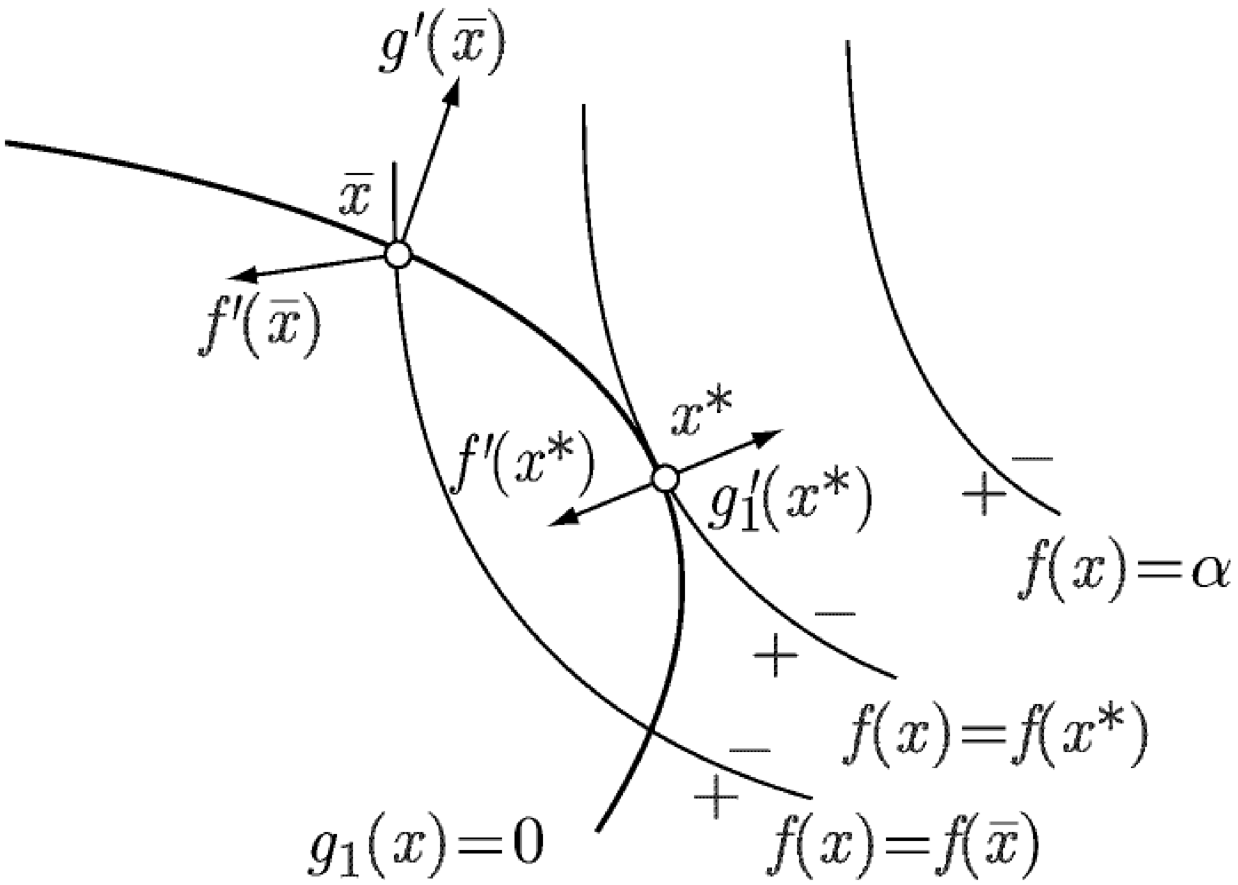
Theorem 6. Suppose that the pair (x^*, λ^*) satisfies the conditions of Theorem 4. If we have that

$$d^2L(x^*, \lambda^*) \geq 0$$

$$\text{for all } \ 0 \neq dx \in T_{x^*}\mathbb{R}^n \text{ such that}$$

$$dg_j(x^*) = 0, j = 1, \dots, m$$

then x^* is a local minimum.



I. Necessary and Sufficient Conditions for Extrema

- ***Equality Constraints. Sufficient conditions.***

$\nabla_x L(x^*, \lambda_0^*, \lambda^*)$	$g_j(x^*)$, $j = 1, \dots, m$	$\lambda_0^* \neq 0, d^2 L(x^*, \lambda^*)$	$dg_j(x^*)$, $j = 1, \dots, m$	Type of critical point
0	0	> 0	0, $dx \neq 0$	Constrained Local Minimum
0	0	< 0	0, $dx \neq 0$	Constrained Local Maximum
0	0	≥ 0	0	Probably Constrained Local Minimum
0	0	≤ 0	0	Probably Constrained Local Maximum
0	0	$= 0$	0	Need additional information
0	0	$\lesseqgtr 0$	0	Not Constrained Extremal Point

I. Necessary and Sufficient Conditions for Extrema

- ***Inequality Constraints. KKT Conditions.***

$$X = \{g_j(x) = 0, j = 1, \dots, m;\ g_j(x) \leq 0, j = m+1, \dots, l\}$$

Definition 2. The constraint $g_j(x) \leq 0$ is called *active at the point x^** if the equality $g_j(x^*) = 0$ is satisfied. If $g_j(x^*) < 0$ then the constraint is called *passive*.

Definition 3. Denote by J_a the set of indices j of the active constraints at the point x^*.

Theorem 7. Let the point $x^* \in X$ be a local minimum of the function $f(x)$ on the set $X \subset \mathbb{R}^n$. Then there exists the number $\lambda_0^* \geq 0$ and the vector $\lambda^* = (\lambda_1^*, \dots, \lambda_l^*)^T$ not equal to zero at the same time such that we have

$$\frac{\partial L(x^*, \lambda_0^*, \lambda^*)}{\partial x_i} = 0, i = 1, \dots, n; \qquad \lambda_j^* \geq 0, j = m+1, \dots, l; \qquad \lambda_j^* g_j(x^*) = 0, j = m+1, \dots, l.$$

Moreover, if $\nabla g_1(x^*), \dots, \nabla g_m(x^*)$ and $\nabla g_j(x^*), j \in J_a$, are linear independent then $\lambda_0^* \neq 0$.

I. Necessary and Sufficient Conditions for Extrema

- ***Inequality Constraints. KKT Conditions.***

Theorem 8. Let the point $x^* \in X$ be a regular (i. e. $\lambda_0^* \neq 0$) local minimum of the function $f(x)$ on the set $X \subset \mathbb{R}^n$ and λ^* be the vector from the previous theorem. Then we have

$$d^2L(x^*, \lambda^*) \geq 0$$

$$\text{for all } dx \in T_{x^*}\mathbb{R}^n \text{ such that}$$

$$dg_j(x^*) = 0, j = 1, \dots, m \text{ and } j \in J_a, \lambda_j^* \geq 0,$$

$$dg_j(x^*) \leq 0, j \in J_a, \lambda_j^* = 0.$$

Theorem 9. Suppose that the pair (x^*, λ^*) satisfies the conditions of Theorem 7 with $\lambda_0^* \neq 0$ and also assume that the total number of constraints-equalities and active constraints-inequalities is equal to n. In this case we have that if $\lambda_j^* > 0$ for all $j \in J_a$ then the point x^* is a local minimum of the function $f(x)$ on the set $X \subset \mathbb{R}^n$.

Theorem 10. Suppose that the pair (x^*, λ^*) satisfies the conditions of the Theorem 5 with $\lambda_0^* \neq 0$. If $d^2L(x^*, \lambda^*) \geq 0$ for all $dx \in T_{x^*}\mathbb{R}^n$ like in Theorem 5 then the point x^* is a local minimum of the function $f(x)$ on the set X.

II. Numerical Unconstrained Optimization

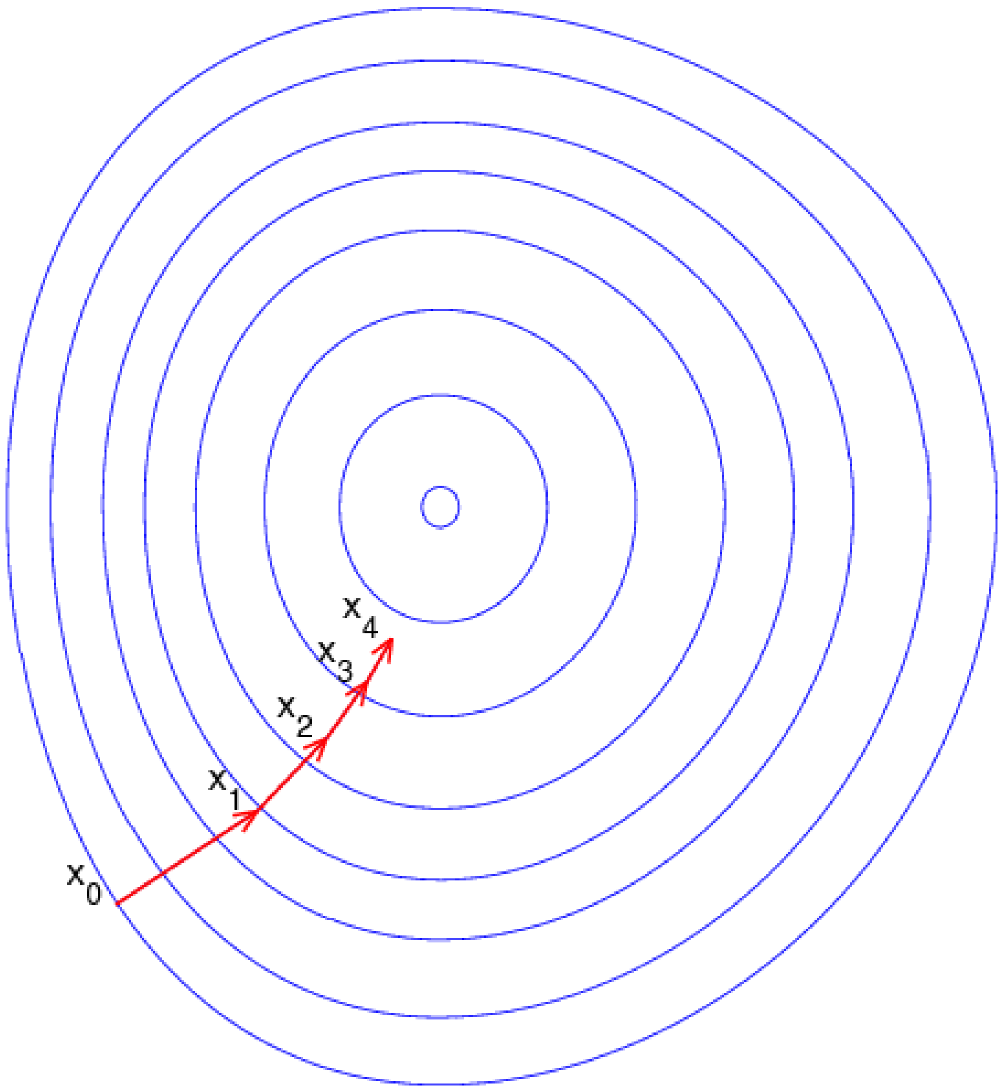
- ***Gradient Descent***

$$f(x) \to min, \qquad x \in \mathbb{R}^n$$

Most of the numerical algorithms for continuous optimization is *iterative* which means that giving initial solution x^0 we generate a sequence of improving approximate solutions $\{x^k\}$ to the problem. The following recurrence relation

$$x^{k+1} = x^k + \alpha_k h^k$$

is often considered where the choice of the vector h^k and the number α_k is based on particular ideas.



II. Numerical Unconstrained Optimization

• *Gradient Descent*

The *Gradient Descent* is an iterative algorithm in which the choice of the vector h^k is based on minimization of linear part of the function $\boldsymbol{f}(\boldsymbol{x})$ around the point x^k, i. e.

$$f(x) - f\left(x^k\right) = \nabla f\left(x^k\right) \cdot \left(x - x^k\right) + o\left(\left\|x - x^k\right\|\right).$$

For fixed step length $\left\|x - x^k\right\|_2 < r$ this linear part achieves minimum value along the anti-gradient direction $-\nabla f\left(x^k\right)$. So we have

$$x^{k+1} = x^k - \alpha_k \nabla f\left(x^k\right).$$

The step size α_k can be chosen such that (we call this *best step size*)

$$f\left(x^k - \alpha_k \nabla f\left(x^k\right)\right) = \min_{\alpha \in \mathbb{R}} f\left(x^k - \alpha \nabla f\left(x^k\right)\right).$$

Remark: For a norm other than the Euclidean norm $\|\cdot\|_2$ the best direction will be different.

II. Numerical Unconstrained Optimization

- ***Newton’s Method***

The *Newton’s Method* is an another iterative algorithm in which the choice of the vector h^k is based on minimization of the quadratic part of the function $\boldsymbol{f}(\boldsymbol{x})$ around the point x^k, i. e.

$$f(x) - f\left(x^k\right) = f_k(x) + o\left(\left\|x - x^k\right\|^2\right),$$

$$f_k(x) = \nabla f\left(x^k\right) \cdot \left(x - x^k\right) + H(f)\left(x^k\right)\left(x - x^k\right) \cdot \left(x - x^k\right).$$

Suppose that the function $f(x)$ is convex then the quadratic part $f_k(x)$ is also convex. One can show that $f_k(x)$ reaches its minimum at the point

$$x^{k+1} = x^k - H(f)\left(x^k\right)^{-1} \nabla f\left(x^k\right).$$

Therefore, in this method we have $h^k = -H(f)\left(x^k\right)^{-1} \nabla f\left(x^k\right),\ \alpha_k = 1.$

II. Numerical Unconstrained Optimization

• *Conjugate Gradient Method*

The previous method requires computation of inverse matrix which can be time consuming operation. The *Conjugate Gradient Method* is the first-order method which avoids this problem and at the same time it converges in most cases as an second-order method. This method is based on the so-called conjugate directions.

Definition 4. Let A be a symmetric positive definite matrix. The vectors h', h'' are called A-*conjugate* if they are non-zero and $\langle Ah', h'' \rangle = 0$. The vectors $h^0, h^1, \ldots, h^k$ are called *mutually* A-*conjugate* if they all are non-zero and we have that $\langle Ah^i, h^j \rangle = 0, i \neq j, 0 \leq i, j \leq k$.

Theorem 11. If the function $f(x)$ is quadratic and $A = H(f)$ then the following approximation

$$x^{k+1} = x^k + \alpha_k h^k, \qquad f\left(x^k + \alpha_k h^k\right) = \min_{\alpha} f\left(x^k + \alpha h^k\right)$$

reaches the minimum point for n steps at maximum.

II. Numerical Unconstrained Optimization

• *Conjugate Gradient Method*

There are different ways of constructing conjugate directions. One of them is based on considering gradients of the function. We choose the vectors h^k as follows

$$h^0 = -\nabla f(x^0), \qquad h^k = -\nabla f\left(x^k\right) + \beta_{k-1} h^{k-1},$$

where the parameter β_{k-1} is chosen such that the vectors h^{k-1}, h^k are conjugate, i. e.

$$0 = \left\langle A h^{k-1}, h^k \right\rangle = -\left\langle A h^{k-1}, \nabla f\left(x^k\right)\right\rangle + \beta_{k-1}\left\langle A h^{k-1}, h^{k-1}\right\rangle,$$

so we have

$$\beta_{k-1} = \frac{\left\langle A h^{k-1}, \nabla f\left(x^k\right)\right\rangle}{\left\langle A h^{k-1}, h^{k-1}\right\rangle}.$$

Theorem 12. If the function $f(x)$ is quadratic then the constructed vectors $h^0, \dots, h^{n-1}$ are $H(f)$-conjugate, so the corresponding stepwise approximation $x^0, x^1, \dots, x^n$ reaches the minimum point for n steps at maximum.

II. Numerical Unconstrained Optimization

- ***Conjugate Gradient Method***

One can rewrite the formula for the coefficient β_{k-1} (for quadratic $f(x)$) in the following way

$$\beta_{k-1} = \frac{\langle Ah^{k-1}, \nabla f(x^k)\rangle}{\langle Ah^{k-1}, h^{k-1}\rangle} = \frac{\langle \nabla f(x^k) - \nabla f(x^{k-1}), \nabla f(x^k)\rangle}{\|\nabla f(x^{k-1})\|^2} = \frac{\|\nabla f(x^k)\|^2}{\|\nabla f(x^{k-1})\|^2}.$$

The right hand side of this equality can be used to adopt the algorithm from the previous slide to minimization of non-quadratic function. Finally, the *Conjugate Gradient Method* is

$$x^{k+1} = x^k + \alpha_k h^k, \qquad f(x^k + \alpha_k h^k) = \min_{\alpha} f(x^k + \alpha h^k),$$

$$h^0 = -\nabla f(x^0), \qquad h^k = -\nabla f(x^k) + \beta_{k-1} h^{k-1},$$

$$\beta_{k-1} = \begin{cases} 0, & \text{if } k = n, 2n, \dots \\ \dfrac{\|\nabla f(x^k)\|^2}{\|\nabla f(x^{k-1})\|^2}, & \text{otherwise.} \end{cases}$$

III. Numerical Constrained Optimization

• *Penalty and Barrier Methods*

Now we proceed to the constrained optimization

$$f(x) \to min, \qquad x \in X \subset \mathbb{R}^n,$$

$$X = \{g_j(x) = 0, j = 1, \dots, m;\ g_j(x) \le 0, j = m + 1, \dots, l\}.$$

Barrier and penalty methods are designed to solve this optimization problem by instead solving a sequence of specially constructed unconstrained optimization problems.

- In a *penalty method*, the feasible region of X is expanded from X to all of $\mathbb{R}^n$, but a large cost or "penalty" is added to the objective function for points that lie outside of the original feasible region X.
- In a *barrier method*, we presume that we are given a point x_0 that lies in the interior of the feasible region X, and we impose a very large cost on feasible points that lie ever closer to the boundary of X, thereby creating a "barrier" to exiting the feasible region.

III. Numerical Constrained Optimization

• *Penalty and Barrier Methods*

Defenition 5. A function $p: \mathbb{R}^n \to \mathbb{R}$ is called a <u>*penalty function*</u> if p satisfies

- $p(x) = 0$ if $g_j(x) \leq 0, j = 1, \dots, m,$ and $g_j(x) = 0, j = m+1, \dots, l.$
- $p(x) > 0$ if $g_j(x) \nleq 0, j = 1, \dots, m,$ or $g_j(x) \neq 0, j = m+1, \dots, l.$

Typical example of penalty function is the following

$$p(x) = \sum_{j=1}^{m} \left|g_j(x)\right|^q + \sum_{j=m+1}^{l} \left(\max\{0, g_j(x)\}\right)^q, \text{where } q \geq 1.$$

Definition 6. A *barrier function* is any function $b: \mathbb{R}^n \to \mathbb{R}$ that satisfies

- $b(x) \geq 0$ for all x that satisfy $g_j(x) < 0, j = 1, \dots, m = l,$ and
- $b(x) \to \infty$ as $\lim_{x} \max_{j}\{g_j(x)\} \to 0.$

Typical example of barrier function is the following function

$$b(x) = \sum_{j=1}^{m} \frac{1}{-\left(g_j(x)\right)^q}, \text{where } q > 0.$$

III. Numerical Constrained Optimization

• *Penalty and Barrier Methods*

The penalty method works as follows

1. Choose an initial solution $x_0 \in \mathbb{R}^n$ which may not satisfy constraints;
2. Choose a suitable penalty function $p(x)$;
3. Choose an increasing sequence of *penalty parameters* $r^1, r^2, \ldots$ (for example, $r^{k+1} = Cr^k$);
4. Giving an approximate solution x_k find a solution x_{k+1} of the following unconstrained problem using the point x_k as a starting point for some numerical method
$$F\left(x, r^{k+1}\right) = f(x) + r^{k+1}p(x),$$
5. If $\left|r^k p(x_k)\right| < \varepsilon$ then the algorithm finishes.

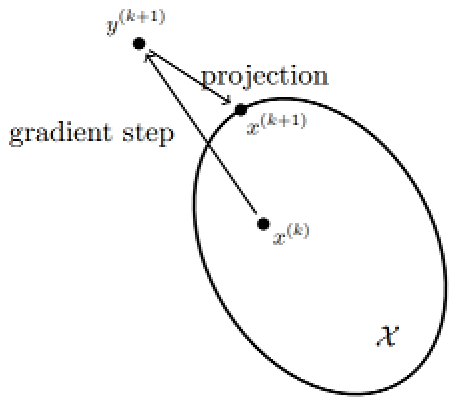
The barrier method works similarly except that we consider a decreasing sequence $r^1, r^2, \ldots$

III. Numerical Constrained Optimization

• *Projected Gradient Method*

Suppose that the set X is closed and convex in $\mathbb{R}^n$. Consider the following algorithm

1. Take any initial solution $x^0 \in \mathbb{R}^n$ (not necessarily from X);
2. For giving x^k consider the point $y^{k+1} = x^k - \alpha_k \nabla f\left(x^k\right)$;
3. Solve the projection problem $\left\|x - y^{k+1}\right\|^2 \to \min,\ x \in X$;
4. Consider the solution x^* of this problem as the next point $x^{k+1} = x^*$.



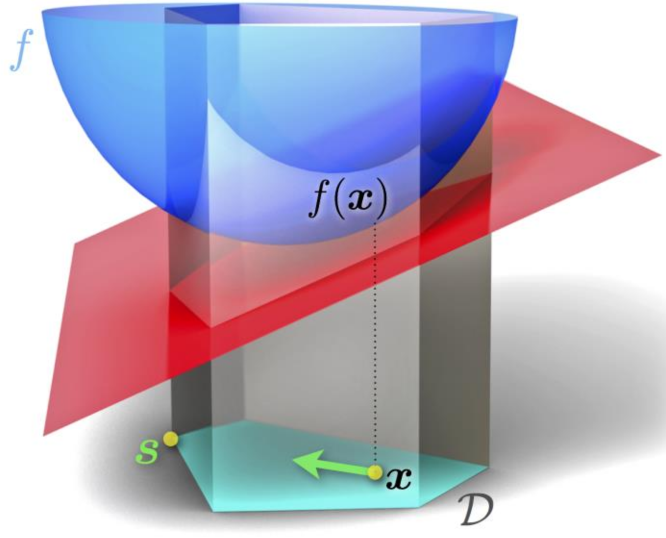
This method can be directly applied only if the set X has a simple structure (for example, X is a ball, parallelepiped, hyperplane or something like that). If X is polyhedron then the step 3 can be efficiently solved by methods of quadratic programming. If X has a more complex structure then apply modifications of the algorithm in which X is replaced by approximate polyhedron.

III. Numerical Constrained Optimization

• *Conditional Gradient Method*

Here also suppose that the set X is closed and convex in $\mathbb{R}^n$. Consider the following method

1. Take an initial solution x^0 from X, $x^0 \in X$;
2. For giving x^k solve the linear optimization problem $\nabla f\left(x^k\right) \cdot \left(x - x^k\right) \to \min,\ x \in X$;
3. Let $\overline{x^k}$ be a solution of this problem, consider a direction vector $h^k = \overline{x^k} - x^k$;
4. Find out the number α_k such that $f\left(x^k + \alpha_k h^k\right) = \min_{0 \le \alpha \le 1} f(x^k + \alpha h^k)$;
5. Consider the next point $x^{k+1} = x^k + \alpha_k h^k$.



As for the previous method this method can be efficiently applied only for the set X with simple structure. You can notice that this method is based on the idea of minimization of linear part of the objective function. Of course, this idea can be generalized. For example, we can minimize linear parts of constraints as well (*Rosen's gradient method*).

Thank you.

Bring digital to every person, home, and organization for a fully connected, intelligent world.

HUAWEI