8 - Solution methods for constrained optimization problems

G. Mastroeni and M. Passacantando

Department of Computer Science, University of Pisa

Optimization Methods and Game Theory
Master of Science in Artificial Intelligence and Data Engineering
University of Pisa – A.Y. 2023/24

Solution methods

Consider the constrained optimization problem defined by

$$\begin{cases} \min \ f(x) \\ g_i(x) \leq 0 & \forall \ i = 1, \dots, m \\ h_j(x) = 0 & \forall \ j = 1, \dots, p \end{cases} \tag{P}$$

Let $X = \{x \in \mathbb{R}^n : g(x) \leq 0, \ h(x) = 0\}$ be the feasible set of (P).

The methods for solving (P) are in general divided in the following classes:

- Primal methods that operate direct on the given problem (P) (e.g., methods of changing the variables, descent direction methods, as projected gradient method or Franke Wolfe method)
- Dual methods, that use the dual of (P), or related properties, (e.g., gradient methods for solving the dual problem, penalty methods)

Problems with linear equality constraints

As an example of a method of changing variables, we consider a problem with linear equality constraints only.

We observe that a constrained problem with linear equality constraints

$$\begin{cases} \min \ f(x) \\ Ax = b \end{cases}$$

where A is $p \times n$ matrix with $\text{rank}(A) = p$, is equivalent to an unconstrained problem:

indeed, write $A = (A_B, A_N)$ with $det(A_B) \neq 0$, where A_B is a $(p \times p)$ matrix. Setting $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$, then $Ax = b$ is equivalent to

$$A_B x_B + A_N x_N = b \quad \Longrightarrow \quad x_B = A_B^{-1}(b - A_N x_N),$$

thus, eliminating the variables x_B,

$$\begin{cases} \min \ f(x) \\ Ax = b \end{cases} \quad \text{is equivalent to} \quad \begin{cases} \min \ f(A_B^{-1}(b - A_N x_N), x_N) \\ x_N \in \mathbb{R}^{n-p} \end{cases}$$

Note that, if f is convex then the previous unconstrained problem is an unconstrained convex problem.

Example. Consider

$$\begin{cases} \min \ x_1^2 + x_2^2 + x_3^2 \\ x_1 + x_3 = 1 \\ x_1 + x_2 - x_3 = 2 \end{cases}$$

Since $x_1 = 1 - x_3$ and $x_2 = 2 - x_1 + x_3 = 1 + 2x_3$, the original constrained problem is equivalent to the following unconstrained problem:

$$\begin{cases} \min \ (1 - x_3)^2 + (1 + 2x_3)^2 + x_3^2 = 6x_3^2 + 2x_3 + 2 \\ x_3 \in \mathbb{R} \end{cases}$$

Therefore, the optimal solution is $x_3 = -1/6$, $x_1 = 7/6$, $x_2 = 2/3$.

Consider a constrained optimization problem

$$\begin{cases} \min\ f(x) \\ g_i(x) \le 0 \qquad \forall\ i = 1, \dots, m \end{cases} \tag{P}$$

Let $X = \{x \in \mathbb{R}^n : g(x) \le 0\}$ the feasible set of (P).

Define the quadratic penalty function

$$p(x) = \sum_{i=1}^{m} \left(\max\{0, g_i(x)\}\right)^2$$

and consider the unconstrained penalized problem

$$\begin{cases} \min\ f(x) + \frac{1}{\varepsilon} p(x) := p_\varepsilon(x) \\ x \in \mathbb{R}^n \end{cases} \tag{P_ε}$$

Note that

$$p_\varepsilon(x) \begin{cases} = f(x) & \text{if } x \in X \\ > f(x) & \text{if } x \notin X \end{cases}$$

Proposition 8.1

1. If f, g_i are continuously differentiable, then p_ε is continuously differentiable and $\nabla p_\varepsilon(x) = \nabla f(x) + \frac{2}{\varepsilon}\sum_{i=1}^{m} \max\{0, g_i(x)\}\nabla g_i(x)$
2. If f and g_i are convex, then p_ε is convex
3. Any (P_ε) is a relaxation of (P), i.e., $v(P_\varepsilon) \leq v(P)$ for any $\varepsilon > 0$
4. If x^*_ε solves (P_ε) and $x^*_\varepsilon \in X$, then x^*_ε is optimal also for (P)
5. If $0 < \varepsilon_2 < \varepsilon_1$, then $v(P_{\varepsilon_1}) \leq v(P_{\varepsilon_2})$

Penalty method

0. Set $\varepsilon_0 > 0$, $\tau \in (0, 1)$, $k = 0$
1. Find an optimal solution x^k of the penalized problem (P_{ε_k})
2. **If** $x^k \in X$ **then** STOP
   else $\varepsilon_{k+1} = \tau\varepsilon_k$, $k = k + 1$ and go to step 1.

Theorem 8.2

- If f is coercive, then the sequence $\{x^k\}$ is bounded and any of its cluster points is an optimal solution of (P).
- If $\{x^k\}$ converges to x^*, then x^* is an optimal solution of (P).
- If $\{x^k\}$ converges to x^* and the gradients of active constraints at x^* are linear independent, then x^* is an optimal solution of (P) and the sequence of vectors $\{\lambda^k\}$ defined as

$$\lambda_i^k := \frac{2}{\varepsilon_k} \max\{0, g_i(x^k)\}, \qquad i = 1, \dots, m$$

converges to a vector λ^* of KKT multipliers associated to x^*.

Remark

Notice that, by Proposition 8.1 (point 5), the sequence of the optimal values $v(P_{\varepsilon_k})$ generated by the penalty method, is increasing.
In fact, if x_ε^* solves (P_ε) and $x_\varepsilon^* \notin X$, then $v(P_\varepsilon) > v(P_{\varepsilon'})$ for any $\varepsilon < \varepsilon'$.

Exercise 8.1

a) Implement in MATLAB the penalty method for solving the problem

$$\begin{cases} \min \ \frac{1}{2} x^{\mathsf{T}} Q x + c^{\mathsf{T}} x \\ Ax \leq b \end{cases}$$

where Q is a positive definite matrix.

b) Run the penalty method with $\tau = 0.1$ and $\varepsilon_0 = 5$ for solving the problem

$$\begin{cases} \min \ \frac{1}{2}(x_1 - 3)^2 + (x_2 - 2)^2 \\ -2x_1 + x_2 \leq 0 \\ x_1 + x_2 \leq 4 \\ -x_2 \leq 0 \end{cases}$$

[Use $\max(Ax - b) < 10^{-6}$ as stopping criterion.]

Matlab commands

```
global Q c A b eps;

Q = [ 1 0 ; 0 2 ] ; c = [ -3 ; -4 ] ;   % data
A = [-2 1 ; 1 1 ; 0 -1 ]; b = [ 0 ; 4 ; 0 ];

tau = 0.1; eps0 = 5; tolerance = 1e-6 ;  % Penalty method
eps = eps0; x = [0;0]; iter = 0; SOL=[];

while true
[x,pval] = fminunc(@p_eps,x);
infeas = max(A*x-b);
SOL=[SOL;iter,eps,x',infeas,pval];
 if infeas < tolerance
  break
  else
  eps = tau*eps;
  iter = iter + 1 ;
 end
end

SOL
```

```
% The penalized function

function v= p_eps(x)

global Q c A b eps;

v = 0.5*x'*Q*x + c'*x ;

for i = 1 : size(A,1)
 v = v + (1/eps)*(max(0,A(i,:)*x-b(i)))^ 2 ;
end

end
```

The Matlab function 'fminunc' (from the Matlab help)

fminunc finds a local minimum of a function of several variables.
[X,FVAL] = fminunc(FUN,X0) starts at X0 and attempts to find a local minimizer X of the function FUN. FUN accepts input X and returns a scalar function value F evaluated at X. X0 can be a scalar, vector or matrix, FVAL is the optimal value of the function FUN.

FUN can be specified using @:

```
  [X,FVAL] = fminunc(@myfun,X0)
```

where myfun is a MATLAB function defined as:

```
  function F = myfun(x)
  F = .........;
```

Consider a convex constrained problem

$$\begin{cases} \min \ f(x) \\ g_i(x) \leq 0 \qquad \forall \ i = 1, \dots, m \end{cases} \tag{P}$$

and define the linear penalty function

$$\widetilde{p}(x) = \sum_{i=1}^{m} \max\{0, g_i(x)\}.$$

Consider the penalized problem

$$\begin{cases} \min \ \tilde{p}_\varepsilon(x) := f(x) + \dfrac{1}{\varepsilon}\, \widetilde{p}(x) \\ x \in \mathbb{R}^n \end{cases} \tag{$\widetilde{P}_\varepsilon$}$$

which is unconstrained, convex and nonsmooth.

Note that

$$\tilde{p}_\varepsilon(x) \begin{cases} = f(x) & \text{if } x \in X \\ > f(x) & \text{if } x \notin X \end{cases}$$

For such penalized problem we do not need a sequence $\varepsilon_k \to 0$ to approximate an optimal solution of (P) (which avoid numerical issues), in fact there exists a suitable ε such that the minimum of $(\widetilde{P}_\varepsilon)$ coincides with the minimum of (P).

Proposition 8.3

Suppose that there exists an optimal solution x^* of (P) and λ^* is a KKT multipliers vector associated to x^*. Then, the sets of optimal solutions of (P) and $(\widetilde{P}_\varepsilon)$ coincide provided that $\varepsilon \in (0, 1/\|\lambda^*\|_\infty)$.

Exact penalty method

0. Set $\varepsilon_0 > 0$, $\tau \in (0,1)$, $k = 0$
1. Find an optimal solution x^k of the penalized problem $(\widetilde{P}_{\varepsilon_k})$
2. **If** $x^k \in X$ **then** STOP
   else $\varepsilon_{k+1} = \tau\varepsilon_k$, $k = k+1$ and go to step 1.

Theorem 8.4

The exact penalty method stops after a finite number of iterations at an optimal solution of (P).

Notice that penalty methods generate a sequence of unfeasible points that approximate an optimal solution of (P).

Exercise 8.2

Run the exact penalty method with $\tau = 0.5$ and $\varepsilon_0 = 4$ for solving the problem

$$\begin{cases} \min \ \frac{1}{2}(x_1 - 3)^2 + (x_2 - 2)^2 \\ -2x_1 + x_2 \leq 0 \\ x_1 + x_2 \leq 4 \\ -x_2 \leq 0 \end{cases}$$

[Use $\max(Ax - b) < 10^{-6}$ as stopping criterion.]

Unlike penalty methods, barrier methods generate a sequence of feasible points that approximate an optimal solution of (P).

Consider

$$\begin{cases} \min \ f(x) \\ g_i(x) \le 0 \quad i = 1, .., m \end{cases} \tag{P}$$

under the following assumptions:

- f, g_i convex and twice continuously differentiable (on an open set containing X)
- there exists an optimal solution (e.g. f is coercive or X is bounded)
- Slater constraint qualification holds: there exists $\bar{x}$ such that

$$g_i(\bar{x}) < 0, \ \forall \ i = 1, \dots, m$$

Hence strong duality holds.

Special cases: linear programming, convex quadratic programming

Logarithmic barrier

On the interior int(X) of the feasible set X, we can approximate the given problem (P) with

$$\begin{cases} \min \ f(x) - \varepsilon \sum_{i=1}^{m} log(-g_i(x)) \\ x \in \text{int}(X) \end{cases}$$

Let $\psi_\varepsilon(x) := f(x) - \varepsilon \sum_{i=1}^{m} log(-g_i(x))$. Setting

$$B(x) := - \sum_{i=1}^{m} log(-g_i(x))$$

then

$$\psi_\varepsilon(x) := f(x) + \varepsilon B(x)$$

Note that, as x tends to the boundary of X, then $\psi_\varepsilon(x) \to +\infty$.

Logarithmic barrier

$B(x)$ is called logarithmic barrier function.

The function $B(x)$ has the following properties:

- $\text{dom}(B) = \text{int}(X)$
- B is convex
- B is smooth with

$$\nabla B(x) = -\sum_{i=1}^{m} \frac{1}{g_i(x)} \nabla g_i(x)$$

$$\nabla^2 B(x) = \sum_{i=1}^{m} \frac{1}{g_i(x)^2} \nabla g_i(x) \nabla g_i(x)^{\mathsf{T}} + \sum_{i=1}^{m} \frac{1}{-g_i(x)} \nabla^2 g_i(x)$$

If x^*_ε is the optimal solution of

$$\begin{cases} \min \ f(x) - \varepsilon \sum_{i=1}^m log(-g_i(x)) \\ x \in \text{int}(X) \end{cases}$$

then

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

Define $\lambda^*_\varepsilon = \left(\frac{\varepsilon}{-g_1(x^*_\varepsilon)}, \dots, \frac{\varepsilon}{-g_m(x^*_\varepsilon)} \right) > 0.$

Consider the Lagrangian function L associated with the given problem (P), $L(x, \lambda) := f(x) + \sum_{i=1}^m \lambda_i g_i(x)$. Then

$$L(x, \lambda^*_\varepsilon) = f(x) + \sum_{i=1}^m (\lambda^*_\varepsilon)_i g_i(x)$$

is convex and $\nabla_x L(x^*_\varepsilon, \lambda^*_\varepsilon) = 0.$

Recall that (P) is a convex problem and strong duality holds, hence

$$v(P) = \max_{\lambda \in \mathbb{R}^m_+} \inf_{x \in \mathbb{R}^n} L(x, \lambda)$$

Consequently,

$$v(P) \geq \min_x L(x, \lambda^*_\varepsilon) = L(x^*_\varepsilon, \lambda^*_\varepsilon).$$

Finally

$$f(x^*_\varepsilon) \geq v(P) \geq L(x^*_\varepsilon, \lambda^*_\varepsilon) = f(x^*_\varepsilon) + \sum_{i=1}^m (\lambda^*_\varepsilon)_i g_i(x^*_\varepsilon) = f(x^*_\varepsilon) - \underbrace{m\,\varepsilon}_{\text{optimality gap}}$$

Remark

Note that:

$$\text{As} \;\; \varepsilon \to 0, \qquad f(x^*_\varepsilon) \to v(P).$$

Interpretation via KKT conditions

The KKT system of the original problem is

$$\begin{cases} \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0 \\ -\lambda_i g_i(x) = 0 \quad i = 1, .., m \\ \lambda \geq 0 \\ g(x) \leq 0 \end{cases}$$

Notice that $(x^*_\varepsilon, \lambda^*_\varepsilon)$ solves the system

$$\begin{cases} \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0 \\ -\lambda_i g_i(x) = \varepsilon, \quad i = 1, .., m \\ \lambda \geq 0 \\ g(x) \leq 0 \end{cases}$$

which is an approximation of the above KKT system.

Logarithmic barrier method

0. Set tolerance $\delta > 0$, $\tau \in (0,1)$ and $\varepsilon_1 > 0$. Choose $x^0 \in \text{int}(X)$, set $k = 1$

1. Find the optimal solution x^k of

$$\begin{cases} \min \ f(x) - \varepsilon_k \sum_{i=1}^{m} log(-g_i(x)) \\ x \in \text{int}(X) \end{cases}$$

using x^{k-1} as starting point

2. **If** $m\,\varepsilon_k < \delta$ **then** STOP
else $\varepsilon_{k+1} = \tau\varepsilon_k$, $k = k+1$ and go to step 1

Choice of starting point

In order to find an initial point $x^0 \in \text{int}(X)$ we can consider the auxiliary problem

$$\begin{cases} \min\limits_{x,s} \ s \\ g_i(x) \leq s, \quad i = 1, .., m \end{cases}$$

- Take any $\tilde{x} \in \mathbb{R}^n$, find $\tilde{s} > \max\limits_{i=1,\dots,m} g_i(\tilde{x})$
  [$(\tilde{x}, \tilde{s})$ is in the interior of the feasible region of the auxiliary problem]
- Find an optimal solution (x^*, s^*) of the auxiliary problem using a barrier method starting from $(\tilde{x}, \tilde{s})$
- If $s^* < 0$ then $x^* \in \text{int}(X)$
  else $\text{int}(X) = \emptyset$

Exercise 8.3.

a) Implement in MATLAB the logarithmic barrier method for solving the problem

$$\begin{cases} \min \ \frac{1}{2} x^{\mathsf{T}} Q x + c^{\mathsf{T}} x \\ Ax \leq b \end{cases}$$

where Q is a positive definite matrix.

b) Run the logarithmic barrier method with $\delta = 10^{-3}$, $\tau = 0.5$, $\varepsilon_1 = 1$ and $x^0 = (1, 1)$ for solving the problem

$$\begin{cases} \min \ \frac{1}{2}(x_1 - 3)^2 + (x_2 - 2)^2 \\ -2x_1 + x_2 \leq 0 \\ x_1 + x_2 \leq 4 \\ -x_2 \leq 0 \end{cases}$$

Matlab commands

```
global Q c A b eps;

Q = [ 1 0 ; 0 2 ] ; c = [ -3 ; -4 ] ;   % data
A = [-2 1 ; 1 1 ; 0 -1 ]; b = [ 0 ; 4 ; 0 ];

tau = 0.5; eps1 = 5; delta = 1e-3 ; x0=[1,1];   % barrier method

eps = eps1 ; m = size(A,1) ;
SOL=[];

while true
 [x,pval] = fminunc(@logbar,x);
  gap = m*eps;
  SOL=[SOL;eps,x',gap,pval];
   if gap < delta
    break
   else
    eps = eps*tau;
   end
end

SOL
```

```
% The penalized function

function v= logbar(x)

global Q c A b eps;

v = 0.5*x'*Q*x + c'*x ;

for i = 1 : length(b)
 v = v - eps*log(b(i)-A(i,:)*x) ;
end

end
```